12-2009

Filtering Noise Caused by Sensor Selection for an Ultra-wide Band Position Tracking System

Danyan Ganjali
Clemson University, dganjal@clemson.edu

FILTERING NOISE CAUSED BY SENSOR SELECTION FOR AN ULTRA-WIDE BAND POSITION TRACKING SYSTEM

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Engineering

---

by
Danyan Ganjali
December 2009

Accepted by:
Dr. Adam W. Hoover, Committee Chair
Dr. Richard R. Brooks
Dr. John N. Gowdy

ABSTRACT

A filter designed to be optimal for alternating multi-sensor tracking systems is presented. This filter can be used on ultra-wide band position tracking systems such as the Ubisense system. In this thesis, a comprehensive evaluation of the accuracy of the Ubisense tracking system in a multi-room building is given. Then the multi-modality of the noise distribution of such a system is shown. A multi-sensor tracking system is then simulated and a Kalman filter is used to filter the measurements. It is shown that the Kalman filter is not optimal for such a system and the need for a filter that addresses the issue of multi-modality is explained. Finally a modified particle filter is designed and is shown to effectively reduce the noise in an alternating multi-sensor position tracking system.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Hoover, for all of the helpful guidance he gave me throughout the Masters' program. I'd also like to thank my committee members, Dr. Gowdy and Dr. Brooks. I would like to dedicate this work to my family, especially my parents, John Ganjali and Zahra Ghasemi. Last but not least, I would like to thank all my friends at Clemon, your support and motivation kept me going through the hard times.

TABLE OF CONTENTS

Page

LIST OF FIGURES

List of Figures (Continued)

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

This thesis considers the problem of noise caused by changing sensor sets for an indoor ultra-wideband (UWB) position tracking system. For outdoor position tracking, the global positioning system (GPS) is well-known. A set of satellites broadcasts signals which are used to solve a lateration problem to determine the position of the receiver. For indoor position tracking, UWB technology can be used to accomplish a similar purpose. A tag broadcasts a UWB pulse to a set of receivers. The receivers estimate the time difference of arrival of the signals to solve a lateration problem to determine the position of the tag. However, although the operating principle of the two systems is the same, the scale of implementation (the Earth versus a single building) causes an important noise problem to arise. When the set of receivers used to perform the lateration calculation changes, this causes the measurement noise to change. For the GPS, this problem happens infrequently due to the size of the coverage and the desired tracking accuracy. For an indoor UWB tracking system, this problem occurs frequently, and affects the desired accuracy.

In a multi-sensor system, there are times that some of the sensors are not available or the system chooses a selected number of sensors to calculate an estimate of a parameter. This can be due to different factors such as the sensor is not able to sense any data, the sensor is out of reach, it is not considered accurate enough by the system or is simply turned off. This can change the dynamics of the system and affect the noise model. In designing a filter to estimate the parameters based on the measurements acquired from the sensors, one should keep in mind which set of sensors are being used, and what noise model is appropriate for that instant of time.

To produce a noise model, the accuracy of a positioning system is often desired to be known. We have all heard a single number summarizing the accuracy of a system in the past. In reality however, things are not as simple as people would expect them to be. More often than not, the accuracy of the system is not the same throughout the trackable area. Some areas can have high accuracies while other areas can suffer from low precision in measuring the position. This is due to various factors that can change in different technologies. However it can be said that for all the positioning systems, the accuracy of the system is not constant because the noise distribution is not

uniform throughout the trackable area.

In this work we first describe the indoor position tracking problem. We discuss several technologies, including UWB, that can be used to build such a system and report their reputed accuracies. We then discuss UWB technology and how it works. This includes a description of a commercially available UWB position tracking system. Such a system was installed and used for the experiments in this thesis. We analyze the performance (accuracy) of this system in order to characterize the noise in its measurements. It is shown that the noise model is not uniform throughout the trackable area and the factors that contribute to this non-uniformity are then studied. This analysis shows the effects of changing sensor sets during tracking. The last part of the thesis includes a filtering method specifically designed for indoor tracking systems. We designed a particle filter that models the noise caused by changing sensor sets, and attempts to filter it out. This filter was implemented on simulated data. Experiments demonstrating its performance are shown.

1.1 Indoor Position Tracking - Review

The broad use of the GPS has turned it into a commodity technology. The importance of the less known short-range and more specifically indoor tracking systems is also becoming more apparent as they have been used in a variety of applications. This type of positioning systems can be used in a wide range of fields such as the military, help and rescue, entertainment, warehousing, assembly lines, athletes training, emergencies and hospitals [18, 12].

Tracking systems can help search and rescue teams and firemen to work more efficiently and avoid walking into hazardous situations flagged by other rescuers. An efficient tracking system can eliminate redundant searching or help the users guide the rescuers to the exact location of the injured ones. In assembly lines, the setting of the machinery, such as the torque applied by drills, can be adjusted automatically to appropriate values for each product, as the tracking system senses the entrance of the tag attached to that product. In gaming players can enjoy a more realistic experience as they immerse into the virtual reality world with the help of a positioning system. Sports teams and athletes can study game replays and positioning analysis using tracking systems to better improve individual and team plays. In warehouses and factories, objects and people can be tracked easier for more efficient productivity or quicker access to tools or professional personnel. In a similar manner in hospitals patients and doctors can be easily tracked to save time that can often save lives.

One application of the tracking systems that has been considered and studied is in the military. Military Operation in Urban Terrain (MOUT) is an ongoing project at the Electrical and Computer Engineering Department of Clemson University and is currently sponsored by the Navy. This project is focused within a single building and with the enemy forces inside the rooms of this building. The soldiers' task is to clear the building and eliminate the enemy. The need for a system that enables tracking soldiers becomes essential and can help eliminate any potential redundant searches of the areas. A map of the building can be made as the soldiers move through the unfamiliar building and if a team member gets wounded, the rescue team can be directed to his location using the map. For the purpose of training, the exact position of each soldier needs to be tracked. For example, what side of the hallway they are on, which wall they are leaning against, or how far apart they are from one another. The accuracy for such a system needs to be in the range of 10 cm in order to have dependable results. A building has been specifically built for the MOUT project in Pendleton, SC. This building is called the Shoothouse.

1.2 Positioning Systems Technologies Available

The accuracy of the tracking system can play a vital role in determining if a certain technology is suitable for an application. It is therefore important to have an understanding of how each of these technologies work and what they are limited to. This section will go over different positioning technologies available while putting more emphasis on indoor tracking systems, and discussing their range of accuracy. There are different approaches that have been taken towards tracking objects in an indoor environment. Some of these technologies are discussed next.

1.2.1 GPS

Global Positioning System (GPS) has shown the benefits and potentials of a positioning system. Although not functional for indoor positioning due to weak signal coverage [23], GPS systems have made tracking systems a well known technology and since some of its techniques are similar to indoor tracking, a brief introduction of the GPS system will be given.

Soon after the lunch of the Russian satellite Sputnik I, it was discovered that the orbit of the satellite could be estimated by analyzing the Doppler shift. It was then suggested that by knowing the position of the satellite, the location of a receiver could be estimated by using this Doppler Shift [17]. This gave birth to the idea of a global positioning system. Officially known as Navstar GPS,

GPS was first developed by the military and was later made available to the public.

The GPS system consists of a constellation of 24 satellites that orbit the earth in about 12 hours. This number is often increased in order to replace the older satellites in orbit with new ones [14]. A GPS receiver can calculate its position in three dimensions from four satellites as shown in Figure 1.1. The receiver's clock determines a psuedo-range $R^{'}$ to each satellite based on the arrival time of the signal. Since the inexpensive clock used by the receiver is not accurate, it introduces an unknown error to the four pseudo-ranges. There are therefore four unknowns, the user's longitude, latitude and altitude as measured from the center of the earth, and the receiver's clock error to bring its clock into synchronization with system time. Solving these four unknowns require four satellites and produces accuracies of about 15 meters [14].

The accuracy of the GPS system can be improved by using Differential GPS (D-PGS), which corrects the error with measured bias errors at a known position. A reference receiver, or base station, computes corrections for each satellite signal [5].

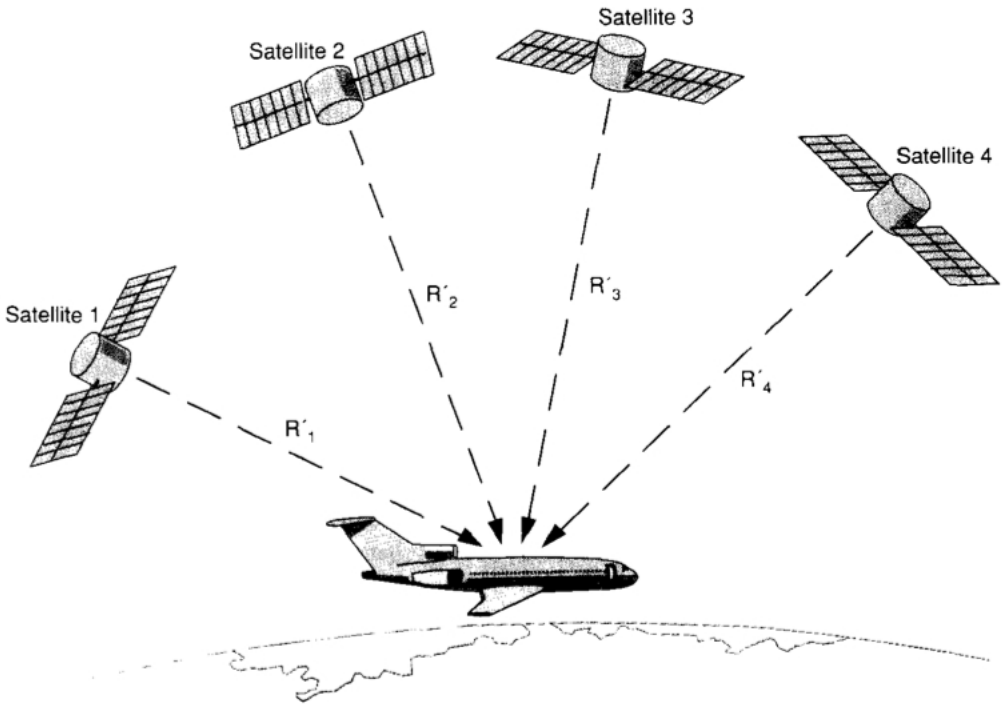


Figure 1.1 Four satellites needed to find the user's position [14]

1.2.2 Radio Frequency Identification (RFID)

A typical RFID system consists of two basic components, RFID scanners and RFID tags. The scanner is able to read the data emitted from RFID tags. There are two types of RFID devices, passive and active. In passive devices, the tags do not use any batteries and they reflect a transmitted signal from the reader and add some information by modulating the reflected signal. They are mainly used to replace the traditional barcode technology. Passive devices have a range between 1-2 meters. Active devices use tags equipped with batteries, which are essentially small tranceivers that actively transmit their ID in reply to an interrogation. The range of active RFID is much larger than passive devices; they can work up to tens of meters. RFID devices can only report the presence of the tag in the area and therefore are not suitable for high accuracy tracking [23].

1.2.3 Wireless Local Area Network (WLAN)

In a WLAN-based tracking system, when a receiver moves in the area, its RF signal strength is reported to a host client. The location is then estimated based on fingerprints in the database which were created during a calibration process [19, 20]. WLAN tracking systems have become very popular in public hotspots and enterprise locations during the past few years. With the dominant use of IEEE 802.11 as the networking protocol, the backbone of a WLAN tracking system is already in place, and incorporating a tracking system in most cases only requires an addition of a piece of software to the host client. WiFi based tracking systems such as Ekahau can utilize the existing Wi-Fi access points installed in the facility and use the network cards already installed on user devices [10].

Although WLAN tracking systems can be installed with minimum hardware, the calibration process requires the user to walk through the entire facility and collect data to build a fingerprint database [19]. Moreover, they are often 2.5 dimensional positioning solutions, meaning that they cannot give 3D coordinates of the position. What they can compute is the 2D coordinates and what floor the receiver is on in the facility [20]. Relatively low accuracy is another disadvantage of such a system. The accuracy of a typical WLAN tracking system is approximately between 3 to 30 meters [23].

1.2.4 UWB

Over the recent years, the UWB has emerged as a promising technology in a variety of fields of study. The nature of this technology enables the UWB signals to pass through walls, making it

Technology	Coverage	Reputed accuracy
GPS	Worldwide (outdoors only)	10 m
Ultrasound	Single room	< 1 m
RFID	Several rooms	-
WLAN (fingerprinting)	Several rooms	3-30 m
Camera	Single room	< 1 m
UWB	Whole Building	< 1 m

Table 1.1 Tracking technologies and their reputed range and accuracy [23, 24]

useful for multi-room buildings such as the Shoothouse. The UWB technology is promising not only for tracking systems, but is also suitable for ultra-fast data transmission, through-wall radar and wireless networking and is expected to be featured in more applications in the coming years [11, 28]. The reported accuracies of the UWB technology is under a meter, and often below 50 *cm*. It is worth having a more in-depth look at this technology since understanding its background played an important role during this project. Section 1.3 discusses this technology with its advantages and design challenges. From this point on, sensors denote receivers that are stationary and receive the UWB signals, and transmitters that are tracked and transmit UWB signals are called tags.

1.2.5 Other Tracking Technologies and Comparison

There are other tracking technologies available on the market. Tracking systems have been built that use technologies such as Bluetooth, ultrasound, cameras and lasers to get an estimate of the position. These technologies are primarily used in single rooms rather than whole buildings and are therefore not discussed here. For a more detailed review of tracking systems and technologies refer to [23].

Table 1.1 summarizes the range of different tracking systems discussed earlier and lists their reputed accuracy.

1.3 Overview of UWB Technology

UWB technology has seen a tremendous increase in interest after the FCC approved it for commercial use in 2002 [13]. The concept behind UWB is based on sending ultrashort pulses, which are usually less than 1 *ns* and have a low duty cycle (typically 1:1000), which translate into low average prime power requirements, ideal for battery-operated equipments [11, 12, 13]. On the spectrum domain, the UWB RF signal is transmitted over a large bandwidth (width >500 MHz). This distinctive nature of UWB, makes it a unique RF signal. Unlike conventional RF signals

which operate on single bands of the radio spectrum, UWB transmits signals over a broad band of frequencies, usually between 3.1 to 10.6 GHz. The presence of lower frequencies in the baseband makes the UWB signals capable of penetrating through walls [13]. The duration of these signals are also much shorter as mentioned earlier, which enables the UWB transmitters to consume less power than conventional RF transmitters. UWB transmitters send pulses sufficiently narrow in time to allow for path resolution at the receiver, avoiding overlap of the pulses which may otherwise combine in a destructive manner and render poor results [13]. Also the short duration of the UWB pulses makes it easier to filter the signal in order to determine which signals are correct and which are produced from multipath, therefore resolving multi-path fading. Moreover UWB systems can be used in proximity of other RF signals without causing interference since the UWB signals use different radio spectrums [13].

UWB technology has certain design issues that need to be addressed. As mentioned before, the broad band of frequencies in the UWB, which include low frequencies, makes a UWB signal capable of going through walls. Therefore the phase information of the signal is equally as important as the magnitude. In narrowband wireless communication however, the quantity of interest has only been the magnitude of insertion loss [27]. While UWB signals suffer attenuation when propagating through walls like other RF signals, they also suffer distortion due to dispersive properties of the walls. Therefore, the properties of the wall or the medium that the UWB signal travels through is important. When a typical RF signal travels through a medium, the entire spectrum travels with the same speed and suffers the same attenuation. But since the complex dielectric constant varies with frequency, the UWB signals behave differently. Due to this fact, different spectral components of the UWB suffer different amount of delay and attenuation [27].

When there is a direct path between a UWB transmitter and receiver, the term line-of-sight (LOS) is used. Similarly when there is no direct line of sight and the signal has to go through objects to get to the receiver, the term no-line-of-sight (NLOS) is used. As stated before, delay and distortion is introduced when a UWB signal travels through an object. Therefore LOS and NLOS conditions produce different signals at the receiver. Figure 1.2 shows a typical impulse response for LOS and NLOS conditions. By knowing the time-of-flight estimated from the delay τ_f, ranging systems can calculate the distance between the transmitter and the receiver. The delay τ_f is the time of the arrival of the first impulse response or leading edge. The range can be calculated by

multiplying τ_f by c, the speed of light. Although UWB systems can successfully isolate multi-path arrivals caused by the signal bouncing off of objects, the response of LOS condition still becomes distorted as shown in Figure 1.2 (a) [13].

Detecting τ_f in NLOS is more troublesome (Figure 1.2 (b)). The leading edge of the signal propagated through walls and objects can be attenuated with respect to reflected signals caused by multipath. At times, the amplitude of the leading edge can be under the limits of signals considered by the system. Even if detected, the leading edge propagating through walls travels slower than it would in free space [13]. This introduces an error in the estimation of each time delay, τ_f, and consequently the range distances and the overall system [6]. For example sheet rock introduces an additional delay of 1.8 ns/m wall for a total range error of 54 cm through 10 walls typically 10 cm thick [13, 27]. In his paper, Muqailbel [27] reports how a UWB signal propagates through walls of different materials, such as drywall, cloth partition, structure wood, wooden door, glass, brick, and concrete block, and how a system should be designed according to what material the building is made out of.

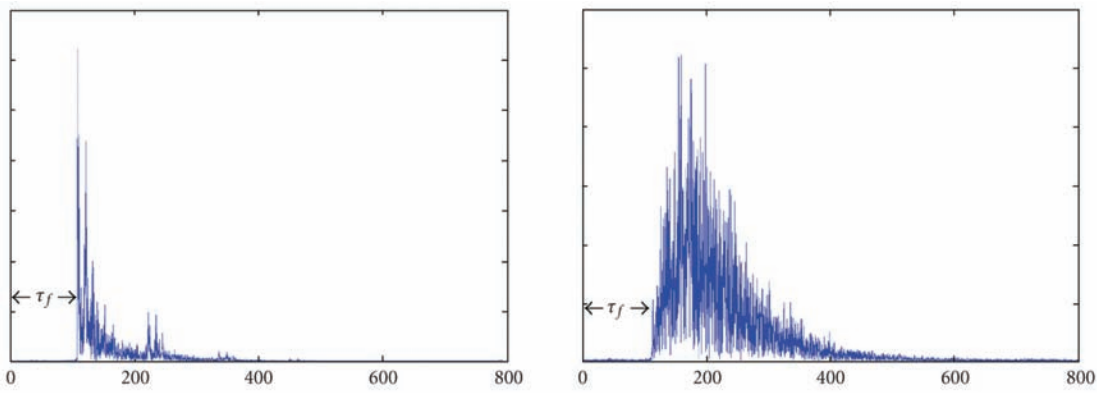


Figure 1.2 Signals at the receiver in LOS and NLOS conditions [13]

1.4 Evaluating A Tracking System

There have been various experiments done by others to evaluate the accuracies of tracking systems. The experiments this paper is focused on are performed on tracking systems that are in indoor environments. Most of the papers found in the literature evaluate the accuracies in laboratories,

where walls and metallic objects are kept away. The accuracies reported by commercial companies tend to match the accuracies obtained in a lab environment. There are however papers that report the accuracy of tracking systems in buildings, although the number of papers that concentrate on UWB-based tracking evaluation is further reduced due to the fact that the UWB technology is relatively new.

Coyle studies the accuracy of the Ubisense tracking system in [4], which is the system also used in this project. The experiments are performed in an empty room without the presence of metal objects. This evaluation is carried out using four Ubisense sensors (receivers) positioned in corners of the room. A grid system made out of squares each with 30 *cm* width, is marked in the room. To test the accuracy, a tag is attached to a user at a height of 120 *cm* and is held stationary in each square for 30 seconds. This is done with the person facing North, West, South and East and the mean and standard deviation for each orientations are computed. The paper reports the error computed for each orientation and states that the accuracies reported by the company are only under optimal conditions. It concludes that the body of a human being could also be a factor effecting performance, along with walls and metallic objects, and that with more sensors in direct line-of-sight, the accuracy increases [4]. The actual position of the tag however can be inaccurate since a it is attached to a person standing on a square. Moreover, the paper does not evaluate the tracking system in the presence of walls and object. In our approach a tripod is used instead of a person which can be positioned accurately on a desired point. This tripod is then positioned throughout the entire building to collect data which is used to produce a comprehensive overview of the accuracy.

In [13] Camillo studies the effect of LOS and NLOS on the accuracy and how the bandwidth and the center frequency should be adjusted depending on the material of the building. The experiment is performed in four buildings each made out of different materials. In each building, twenty five points are taken and the estimated positions are compared with the ground-truth positions obtained by a laser range finder [13]. The paper however does not provide a comprehensive study of the accuracy in these buildings.

Performance evaluation for WLAN systems can use the grid system already created from the calibration process to compare the live positions to the calibrated positions [19, 20]. The challenge in evaluating such a system is to build an accurate grid system for ground-truth measurements in a

multi-story building. In a UWB tracking systems however, the calibration techniques are different and a fingerprints database is not made previously, therefore carefully evaluating the system means creating a grid system in the building.

Other reports such as the two performed at Disaster City at Texas A&M university and at National Institute of Standards and Technology (NIST) reported in [20], often sum up the accuracy in a single number or a range, which does not fully demonstrate the system's performance. In order to perform a comprehensive evaluation, a novel way to create an accurate grid system is presented in the next chapter. The accuracy of a tracking system (Ubisense) is then computed by comparing the reported measurements and the ground-truth position obtained from the grid system.

1.5 Filtering

As stated earlier, there are various sources of error that can impact the performance of a UWB indoor tracking system. To filter out the noise and to improve the measurements reported by the tracking system, we consider two widely known filters, the Kalman filter and the particle filter. The next two sections gives a brief introduction to each of these two filters.

1.5.1 Kalman Filter

The Kalman filter is an optimal recursive data processing algorithm [26] first proposed by R. E. Kalman. It estimates the state of a parameter based on all measurements data available, with the prior knowledge of the system and the measuring devices used, in such a way that the statistical error is minimized [26]. The word recursive means that the Kalman filter does not require the previous states to be saved and reprocessed every time a measurement is taken [26]. In other words, to calculate the next state, only the current state and measurements are needed.

The Kalman filter has some limitations. It is only optimal with systems that can be described by linear models and in which the system and measurement noises are zero-mean Gaussian distributions [26]. The Extended Kalman filter (EKF) is a modified version of the Kalman filter that is used to work with non-linear systems [34]. For a thorough explanation of the Kalman filter refer to Chapter 2 of the book written by Brookner [2].

1.5.2 Particle Filter

The particle filter was first introduced in 1993 in [16]. It is an algorithm for implementing recursive Bayesian filtering. The particle filter can be used on nonlinear and non-Gaussian problems and its performance is superior to the Extended Kalman filter [7]. The central idea behind this filter is to represent the PDF as a set of samples instead of a function over the state space [16]. With enough samples used, they effectively provide an equivalent representation of the PDF. With the increase of the sample size however, the amount of computation increases as well. But with the advances of computers over the past years, the speed at which these calculations are made has increased. The potential and popularity of the particle filter has therefore multiplied [9].

As stated above, a distribution in a particle filter is approximated by discrete random measures defined by the particles and their weights,

$$p(x) = \sum_{m=1}^{M} w^{(m)} \delta(x - x^{(m)}) \tag{1.1}$$

where $x^{(m)}$ are the particles and $w^{(m)}$ are their weights [9].

Figure 1.3 demonstrates the three different steps that the particle filter goes through. At first the M particles are generated. Then at time t all the particles are weighted according to the distributions of interest. Note that these distributions do not have to be zero-mean Gaussian; in fact they can be any distribution. The distributions can also be different every time this process is performed. The size of the particle in the figure reflects the size of the weight of that particle. Particles with larger weights have a higher probability of surviving. With this happening over time, all particles except only a few, would be assigned negligible weights. This degeneration significantly deteriorates the performance of the filter. To prevent this, smaller particles are removed, and larger particles are replicated a number of times according to their weights. This algorithm is called *resampling*, which is also demonstrated in Figure 1.3. The particles are then normalized and the three steps are performed again [9]. For a more detailed introduction and explanation of the particle filter please refer to [9].

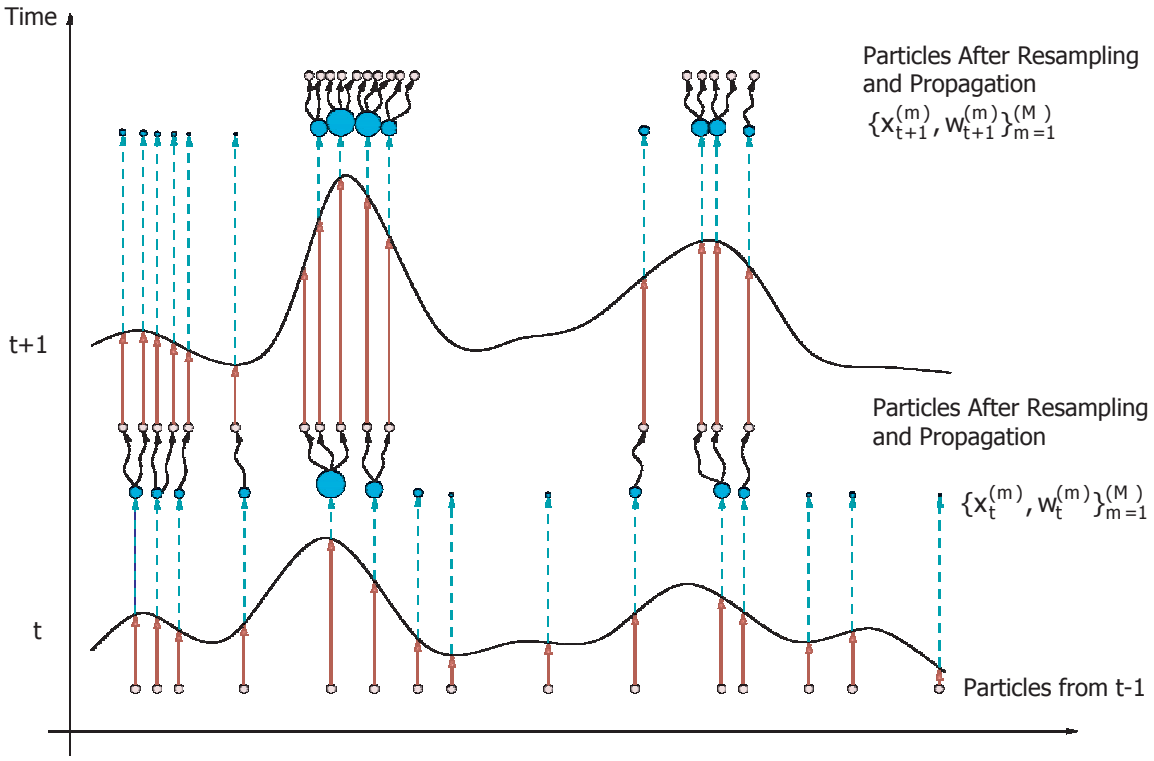


Figure 1.3 Particle filter steps [9]

CHAPTER 2

METHODS

This chapter discusses the different steps of the project and the methods that were used to approach each of these steps. The results are then summarized in the next chapter.

2.1 Selecting a UWB Positioning System

One of the tasks prior to this research was to select a positioning system for the MOUT project. Although a camera system had been used to track soldiers inside the Shoothouse (Figure 2.1), the tracking capabilities were limited and a need for a more precise and reliable positioning system was essential. As discussed before, a UWB system would offer higher precision and reliability. Therefore a research on the different UWB transceivers available on the market was performed. It was first decided to develop a tracking system using these UWB transceivers, which are sold by several different companies (i.e, Time Domain). This is, however, an extensive study topic and would require a lengthy research time in order to produce a precise tracking system. The second option was to purchase a complete UWB indoor positioning system.

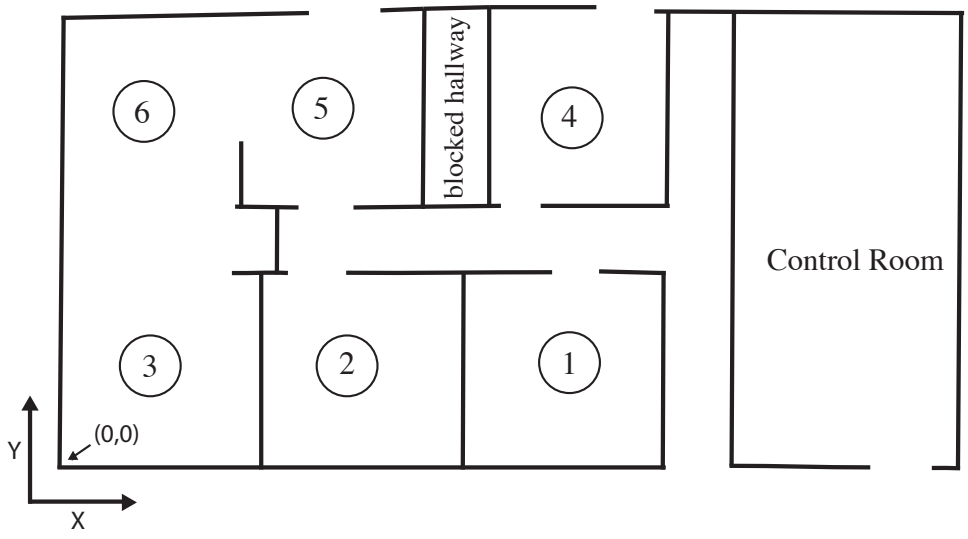


Figure 2.1 Map of the Shoothouse with the rooms numbered

Figure 2.2 The MOUT project Shoothouse

2.1.1 UWB Positioning Systems Available on the Market

There are a handful of companies working on tracking systems using the UWB technology. Several companies developing UWB-based tracking systems were contacted, but at the time only two had a full working system ready to be installed. Systems using competing technologies were also considered in order to compare against the UWB system. Table 2.1 shows the different companies considered and the technology/algorithm that they employ. The accuracies reported by the companies in this table are under ideal conditions.

Company	**Reported accuracy**	**Algorithm**	**Technology**
Time Domain	3' to 1'	TDOA	UWB
Ubisense	15 *cm* (6") in 3D	AOA and TDOA	UWB
MultiSpectral Solutions	30 *cm* (12")	TOF	UWB
Ekahau	Several feet	Fingerprinting	Wi-fi

Table 2.1 Tracking companies considered

After contacting the companies and reviewing their systems, it was decided to purchase the Ubisense tracking system. The next section gives an overview of what the Ubisense tracking systems offers and how it operates.

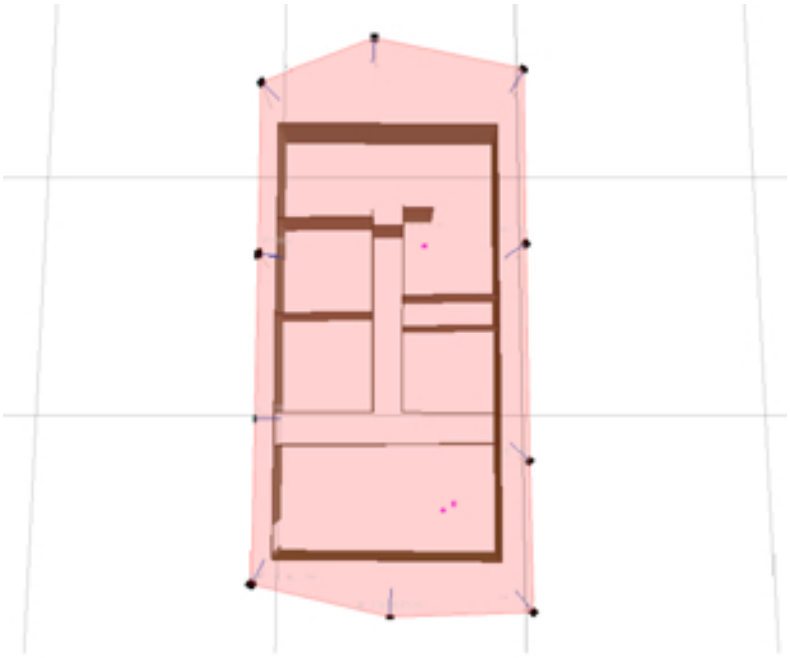


Figure 2.3 UWB Tracking System Setup (taken from Ubisense program)

2.2 Ubisense Positioning System

Based in Cambridge, England, Ubisense Inc. offers real-time tracking systems suited for indoor environments. The system was purchased and installed with the help of a technician sent by the Ubisense company. Figure 2.3 shows the Ubisense sensor configuration for the Shoothouse. The sensors, which are essentially transceivers, are placed around the building. The trackable area is then the region within the sensors. The positioning accuracy observed is not uniform throughout the Shoothouse, which will be discussed in the coming sections. This is due to the factors mentioned earlier, which cause the performance to be noisier than under ideal conditions. The next sections discuss the different components used in the Ubisense tracking system.

2.2.1 Ubisense system setup

The Ubisense tracking system is made out of several different components. These include, sensors, tags, a central router, CAT5 networking cables, installation hardware, a host PC, and the host PC software. The following sections give a brief overview of each component and what role it plays in the Ubisense tracking system.

2.2.1.1 Ubisense sensors

Ubisense sensors (7000 Series) are UWB measurement devices that contain an array of antennas and UWB receivers. The sensors detect UWB pulses that are transmitted from the tags. They also communicate with the tags over a conventional 2.4Ghz radio, which can be used to manage the tag's update rate, sleep mode and other functions [33]. The sensors can be powered either through CAT5 cables by power-over-ethernet from a central router, or by typical DC adapters. The Shoothouse is equipped with ten Ubisense sensors. Each sensor is connected to the existing railing attached to the ceiling with the use of two clamps, which were purchased from a hardware store (Figure 2.4). After securing the sensors in place, each sensor is positioned horizontally level to the ground with the help of a small level tool shown in Figure 2.5. This ensures that the roll angles are zero, which makes the calibration process and the calculations easier.

Figure 2.4 Installed Ubisense sensor

Each sensor in the Ubisense system produces time difference of arrival (TDOA) and angle of arrival (AOA). These set of measurements are used to calculate the estimate of the location of a tag. With knowing the TDOA measurements, the distance of the tag can be calculated using the time of flight. The angle of arrival on the other hand can be used to estimate the direction of the tag with respect to the North or South direction. This direction is a line on a 2D plane. It will be explained

later on in the chapter that with the use of TDOA and AOA from two sensors, the position of a tag can be estimated.

Figure 2.5 Positioning a Ubisense sensor horizontally using a level tool

2.2.1.2 Ubisense Tags

Ubisense tags (Series 7000 Compact Tag) are essentially UWB transmitters that transmit UWB pulses in the band of 6-8GHz. The pulses can be generated at frequencies between 0.0025 Hz up to 33.75 Hz. This rate can be controlled by the user over a conventional RF signal operating at 2.4 Ghz from the sensors. The tags are powered by a 3V coin cell battery. With the tag sending pulses at every 3 seconds, the battery is advertised to last over four years [32]. This is due to the UWB low power consumption discussed in the previous chapter.

2.2.1.3 Router

The job of the router is to relay the data from the sensors to a host machine for final calculations. In the case of the Shoothouse, it also provides power-over-ethernet to all the sensors through CAT5 cables. A router with such capability was selected and installed high above ground and in the middle of the Shoothouse, which made running the cables from the sensors easier. Figure 2.6 shows the installed router.

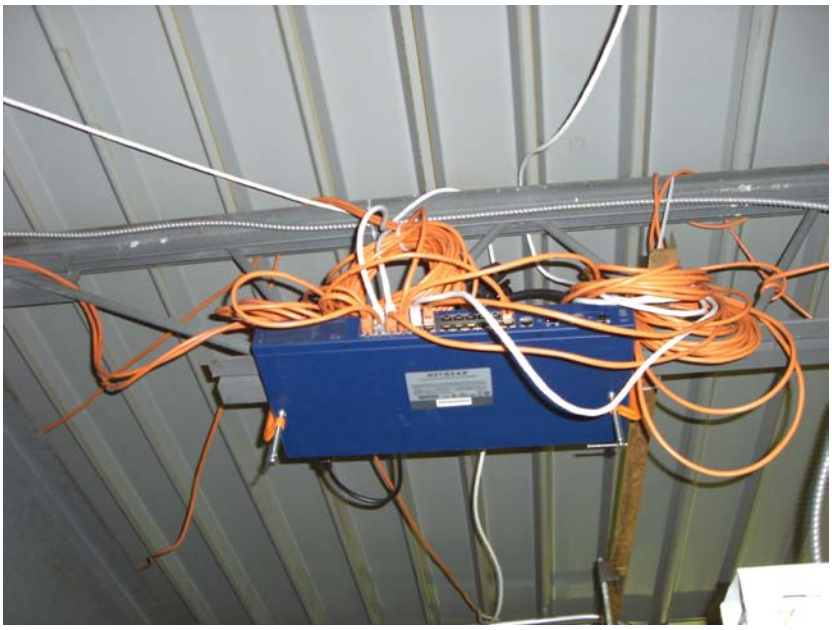


Figure 2.6 Installed router for the Ubisense tracking system

2.2.1.4 Software

The software included in the Ubisense tracking system consists of programs that read the license files and let the user manage the entire system from a host PC. Maps of the building can be precisely drawn and the user is able to define zones on the map where tags are activated or deactivated. The update rate of the tags can also be changed, either to a constant rate, or to a rate that changes depending on the velocity of the moving tag. The calibration process discussed in the next section is also performed on one of the programs in the package.

2.2.2 The Calibration Process

Prior to calibrating the system, the position of all the sensors are manually input into the system. These positions are calculated by the method of lateration. First two known points are picked in the Shoothouse, then with the use of a laser range finder, the distance of these two points to a sensor is found. By knowing the distance of the sensor from each of the two known points, one can find the x and y coordinates of sensor. The z coordinate of the sensor is simply measured by hand.

The next step is to place a tag in a known position in the Shoothouse and start the calibration process. The calibration process is started from the host PC. During the calibration process, the pitch and yaw of each sensor is calculated; the roll is zero as described earlier. The biases in

Figure 2.7 Finding the coordinates of a Ubisense sensor using a laser range finder

estimation of the AOA and the TDOA of each sensor is also corrected with reference to the tag positioned at the known location. The cable delays between each sensor is another parameter that is calculated during the calibration process.

2.2.3 The Ubisense Tracking System Setup

Figure 2.8 shows the Ubisense tracking system setup. Each of the sensors are connected to the router/switch by CAT5 cables, and from there a single cable connects the system to the host PC. The sensors are connected to each other in daisy chain configuration, or all are connected to a sensor selected as the master sensors as shown in the figure. The calculations made on each sensor are forwarded to the master sensor which finalizes the calculation with the use of the time delays, and then forwards everything to the host PC via the router.

2.3 Accuracy of the Ubisense System

After purchasing the Ubisense tracking system and installing it, it was noticed that there are factors that can affect the performance of the tracking system. Although UWB signals can pass through some walls, the tracking system behaves best when its sensors are in direct line-of-sight of the tags. The presence of metal throughout the building is also another issue. As stated in the

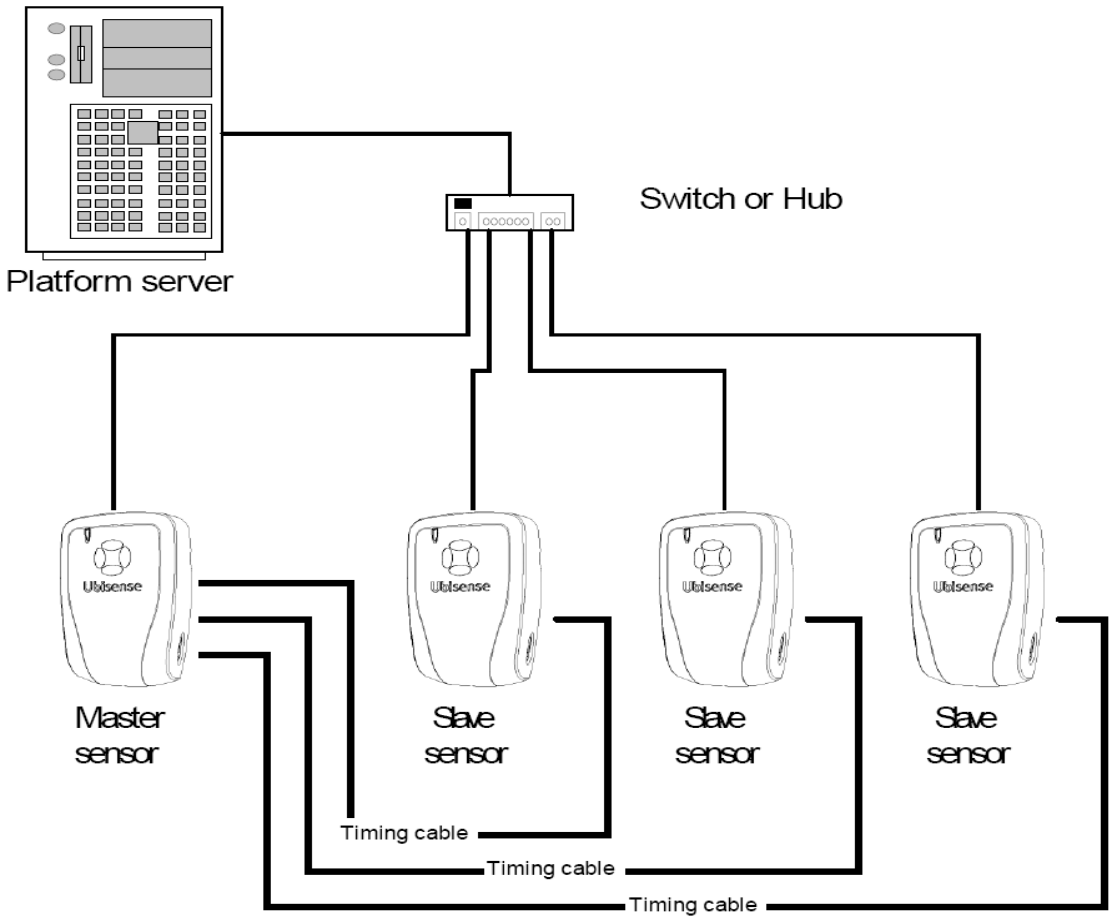


Figure 2.8 The Ubisense system setup [22]

previous chapter, the UWB technology is said to work well around metals due to the fact that the duration of the pulses are very short. This minimizes the multipath interference caused by signals bouncing from metal objects. This however does not mean that metal does not reduce the accuracy of the system. Most building materials including wall studs, AC ducts and roofing material are made of or include metal, which affects the performance of the system.

Since the performance of the Ubisense system was not uniform throughout the Shoothouse, the need for a comprehensive evaluation was apparent. There have been efforts in previous literature to find the accuracy of a tracking system in a controlled laboratory environment as described in the previous chapter. The importance of having reliable ground-truth values and an accurate grid system was also explained. This grid system is then used to collect measurements to compute the

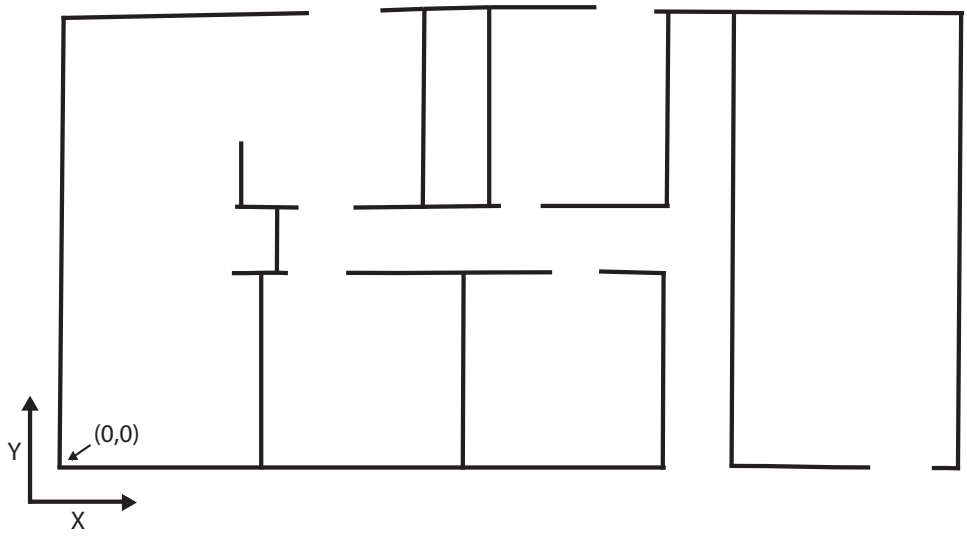


Figure 2.9 The Ubisense system and the ground truth coordinate System

accuracy of the Ubisense system. The next section describes the method for creating an accurate grid system for a multi-room environment.

2.3.1 Acquiring Ground Truth Readings

In order to calculate the accuracy of the measurements, it is important to be able to compare the measurements to a set of ground-truth values. This requires a system in which the actual coordinates of a point can be measured in the trackable area. This section describes an efficient yet simple procedure that can be used in any building ranging from single rooms to multi-rooms.

The first step in setting up this coordinate system is to pick an origin. The origin can theoretically be selected anywhere in the building, but to simplify things the origin of this new coordinate system was picked to be the same as the Ubisense tracking system. This eliminates any transformation from one coordinate to another, and the calculation of the accuracy on a single axis becomes a simple subtraction. Figure 2.9 shows this coordinate system.

To help setup an accurate coordinate system, the Stanley S2 Laser Level Square (Model 77-188) [8] was used. This device, shown in Figure 2.10, projects two accurate perpendicular laser lines on the floor, which makes it ideal for drawing the coordinate system axis on the floor. Once the origin has been picked, the Stanley Laser Square is used to create two perpendicular lines that start/meet

Figure 2.10 Stanley Laser Square used to create the ground truth coordinate system [8]

at this origin point. These two lines are then marked in increments of 25 cm (this can be changed to lower or higher increments). Any point in the same room as the origin can now be spotted with the use of two Stanley Laser Squares. As an example, let us find the point $A(x,y)$=(2 m, 3 m). The first laser device is place on point (2 m, 0 m); this point has already been marked, therefore easily found. The device is then rotated until one of the perpendicular laser lines falls exactly on the line that goes through the origin, while the other laser line goes towards point A. The second laser device is places at point (0, 3 m) in a similar manner just explained. The lines from the two laser devices now intersect at two points, one of them the origin and the other one point A, which is (2 m, 3 m).

In order to find a point in the other rooms of the Shoothouse, the coordinate system needs to be extended to all the rooms. Based on the procedure explained earlier, to find a point in the coordinate system, we need to have two known marked perpendicular lines in a room. By having these two known lines, the two laser devices can be used to find the location of a given point. To extend the coordinate system, the Stanley Laser Square device is again very helpful. To draw a line in a room, the laser device is placed on a last known drawn line in such a way that one of the laser lines is aligned with the marked line, while the other extends to the unmarked room. This laser line is then marked down with a permanent marker drawn on duct tape that is laid down. Now that we have one marked line in a room, the second line perpendicular to the first is simply found using the laser square. Figure 2.11 shows the completed process for all the rooms with the lines carefully drawn.

The accuracy of the laser device claimed by the company is at most half an inch for every 30 feet.

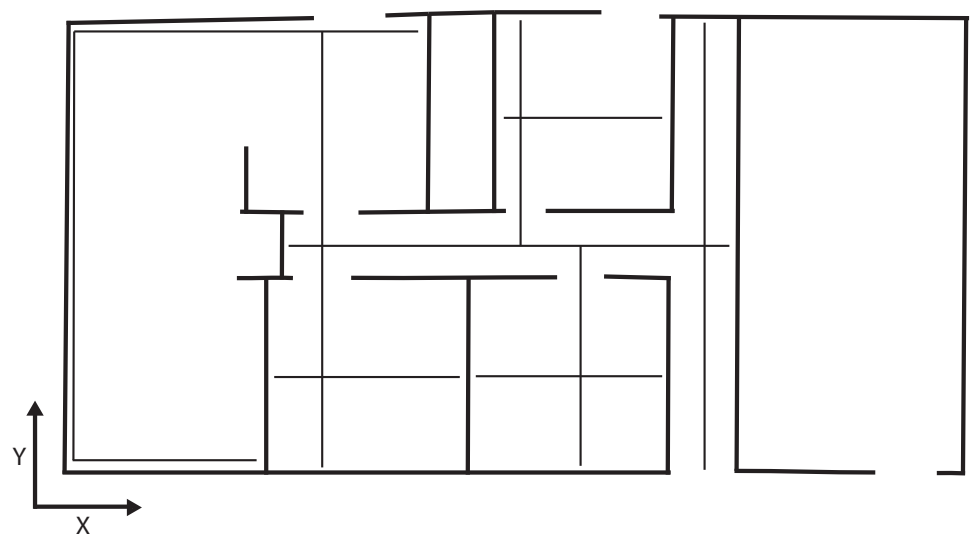


Figure 2.11 Perpendicular lines extended to each room

This means that the projected perpendicular lines are half an inch off for every 30 feet. Since in the case of the Shoothouse the lines do not extend beyond 30 feet, this accuracy was acceptable. In fact, after doing some tests, it was found that the overall accuracy of the system we used was in the range of millimeters. One of the tests done was to choose two different paths to draw the perpendicular lines, and compare the two in the last room. The first path chosen was from room to room, while the second path was to go around the building in some areas and come in the building through the doors. It was noted that the lines meeting at the last room were only off by few millimeters. With a grid system that was accurate down to millimeters, the ground-truth positions were obtained with confidence.

2.3.2 Measuring the Accuracy of The Ubisense System

There are two different approaches that were considered in order to calculate the accuracy of the positioning system. First when the tag (object) is stationary, and second, when the tag is moving. Each of these categories requires a different test for the accuracy, which will be discussed next.

2.3.2.1 Accuracy of A Stationary Object

The accuracy of a stationary object is calculated by comparing the UWB positioning system measurements to the actual position of the object. In order to get the measured position by the

Ubisense system, the tag is placed on top of a tripod, as shown in Figure 2.12, and 100 measurements are recorded. To collect the data a C program is written that connects to the Ubisense tracking system through a UDP socket. The program then stores the measurements in a file. The stored data include the x, y and z coordinates along with the time stamp of each of the 100 samples. The average of these measurements is computed in each of the axis to calculate the estimated position of the tag. They are denoted as X_m, Y_m and Z_m.

Figure 2.12 A Ubisense tag on a tripod used to record stationary readings

Once X_m, Y_m and Z_m are known, they can be compared to the actual coordinates of the tag which can be obtained from the lines available in each of the rooms (refer to the previous section). The actual coordinates are named X_a, Y_a and Z_a. Then the accuracy A in each direction is:

$$A_x = |X_m - X_a| \tag{2.1}$$

$$A_y = |Y_m - Y_a| \tag{2.2}$$

$$A_z = |Z_m - Z_a| \tag{2.3}$$

The overall accuracy in 3-dimensional coordinate is

$$R = \sqrt{(A_x^2 + A_y^2 + A_z^2)} \tag{2.4}$$

To see if the accuracy is consistent in the entire Shoothouse, this process needs to be performed throughout the entire trackable area. The tripod is placed around the Shoothouse at increments of 25 *cm*. The R values for each points are then plotted in 3D using the *surf* command in Matlab. The results are reported in the next chapter.

2.3.2.2 Accuracy of a Moving Object

The accuracy of a moving tag can be measured by recording the position of a tag placed on a motorized system that moves the tag at a constant velocity on a straight and known line. The measurements can then be compared with the actual locations of the tag based on the tag's velocity and the time of measurement. This part of the project however was not completed and is left for future research.

2.4 Distance and location calculation algorithms

There are different RF techniques that are used to calculate the position. They include angle of arrival (AOA), received signal strength (RSS), time of arrival (TOA) and time difference of arrival (TDOA). Each of these techniques have their own merits and draw backs under cost and complexity constraints [1]. In AOA the angle of a node is measured relative to a reference node. With two angles known, position in two dimensional space can be computed [1]. RSS techniques measure the signal energy received and use triangulation to locate the position. With three reference nodes, the two dimensional location of a node can be found [1]. The TDOA uses the difference in time the signal was received by two receivers to estimate the position of a transmitter. With two receivers on hand, the position of the transmitter is placed on a hyperboloid and with the third receiver added, a unique position can be estimated using a least squares iterative methods or non-iterative methods such as the one proposed in [3] and [29] [18, 35, 37]. The TOA technique uses the absolute time that the signal was received. Dealing with the absolute time of the arrival however can be a major source of error in indoor positioning systems. This is due to the fact that calculating short distances travelled at the speed of light needs a very accurate clock, since a small inaccuracy in time calculated causes a big error in the distant estimated. When using the TOA, the position of the transmitter

can be located on a circle around the receiver. With three receivers used, trilateration techniques can be employed to calculate a unique location for the transmitter [15, 23].

The Ubisense system uses an algorithm that combines AOA and TDOA [31]. The advantage of using TDOA in conjunction with AOA is that a location can be determined from just two sensors, decreasing the required sensor density over systems that just use TDOA [23]. In order to simplify the simulation and the illustrations, this paper focuses on the TOA algorithm and the trilateration technique on a 2-dimensional system. However, all the methods shown can be applied to TDOA and hyperbolic positioning systems.

2.4.1 Lateration

As stated earlier, the lateration (trilateration in this case) technique uses the TOA at three receivers to estimate the position of a transmitter. Figure 2.13 illustrates this technique for an ideal system where the TOAs are accurate without any noise. *T1*, *T2* and *T3* correspond to the center of the receivers with *r1*, *r2* and *r3* being the distances from the transmitter that were measured using TOA. Based on the assumption made, the TOA measurements provide the exact distance between a transmitter and a receiver. The position calculated is therefore the true location of the transmitter. By knowing the distance to the transmitter (distance r shown in the figure), the position of a transmitter can be located on a circle drawn centering at the receiver as shown in the figure. The intersection of these three circles is the location of the transmitter [15].

When the TOA measurements are not accurate, which is the case for a system in the real world, the true position of the transmitter does not exactly lie on the circles created around the transmitter. The result of this inaccuracy is that the three circles created for trilateration do not intersect at a common point. The estimated position of the transmitter is then calculated using techniques such as the least squares fit. Figure 2.14 shows the estimated position of a transmitter when the TOA measurements are not accurate.

2.4.2 Solving the Lateration Problem

In a typical positioning system such as the GPS, each sensor relays information such as the pseudorange and the clock offset to the host machine. The pseudorange is the approximate distance between each sensor and the tracking object calculated by multiplying the time difference with the

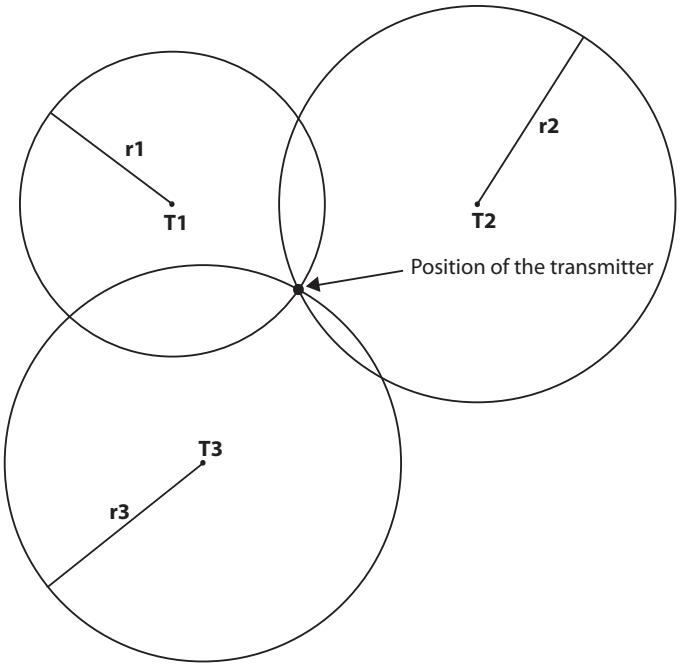


Figure 2.13 Trilateration for an ideal system

speed of light. The clock offset is the the error in the clock of the sensor relative to a reference clock. The following equations are taken from [30].

Let there be j receivers, then pseudorange of each receiver can be written in terms of $\rho^{(j)}$, the actual range, $\delta^{(j)}$, the receiver's clock offset (in GPS it is sent by each satellite), and δ_R, the receiver's unknown clock offset.

$$P^{(j)} = \rho^{(j)} + c\delta^{(j)} - c\delta_R \tag{2.5}$$

The actual range can be written in terms of $(X^{(j)}, Y^{(j)})$, the coordinates of the sensor, and (X, Y), the position of the receiver:

$$\rho^{(j)} = \sqrt{(X^{(j)} - X)^2 + (Y^{(j)} - Y)^2)} \tag{2.6}$$

Rearranging the equation so that the known parts are on the left side and the unknowns are on the right, will produce the following:

$$P^{(j)} - c\delta^{(j)} = \sqrt{(X^{(j)} - X)^2 + (Y^{(j)} - Y)^2)} - c\delta_R \tag{2.7}$$

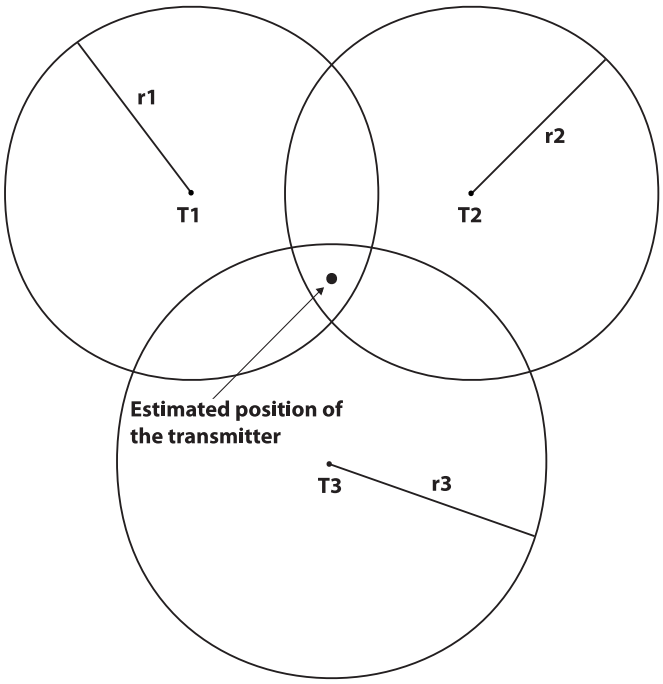


Figure 2.14 Trilateration for a system in the presence of noise

Since the above equation has three unknowns, three equations produced from three sensors are needed to estimate the position in 2D space. This means that for a typical GPS system which finds the position of a target in 3D space, at least four satellites are needed to solve for four equations.

Using the Newton-Raphson method, one can solve the set of equations above.

$$\rho^{(j)} = \rho_0^{(j)} + \frac{\partial \rho^{(j)}}{\partial X}|_{X_0,Y_0}\Delta X + \frac{\partial \rho^{(j)}}{\partial Y}|_{X_0,Y_0}\Delta Y \tag{2.8}$$

where $\rho_0^{(j)}$ is the initial guess for the actual range at (X_0, Y_0).

$$P^{(j)} - c\delta^{(j)} - \rho_0^{(j)} = \frac{\partial \rho^{(j)}}{\partial X}|_{X_0,Y_0}\Delta X + \frac{\partial \rho^{(j)}}{\partial Y}|_{X_0,Y_0}\Delta Y - c\delta_R \tag{2.9}$$

and the derivatives calculated at the initial guess are:

$$\frac{\partial \rho^{(j)}}{\partial X}|_{X_0,Y_0} = \frac{-(X^{(j)} - X_0)}{\rho^{(j)}(X_0, Y_0)} \tag{2.10}$$

$$\frac{\partial \rho^{(j)}}{\partial Y}|_{X_0,Y_0} = \frac{-(Y^{(j)} - Y_0)}{\rho^{(j)}(X_0, Y_0)} \tag{2.11}$$

To solve the above equation, it is written in matrix format. Then it can be solved by multiplying two matrices.

$$l = \begin{pmatrix} P^{(1)} - c\delta^{(1)} - \rho_0^{(1)} \\ P^{(2)} - c\delta^{(2)} - \rho_0^{(2)} \\ P^{(3)} - c\delta^{(3)} - \rho_0^{(3)} \end{pmatrix} \tag{2.12}$$

$$A = \begin{pmatrix} \frac{\partial \rho^{(1)}}{\partial X}|_{X_0,Y_0} & \frac{\partial \rho^{(1)}}{\partial Y}|_{X_0,Y_0} & -1 \\ \frac{\partial \rho^{(2)}}{\partial X}|_{X_0,Y_0} & \frac{\partial \rho^{(2)}}{\partial Y}|_{X_0,Y_0} & -1 \\ \frac{\partial \rho^{(3)}}{\partial X}|_{X_0,Y_0} & \frac{\partial \rho^{(3)}}{\partial Y}|_{X_0,Y_0} & -1 \end{pmatrix} \tag{2.13}$$

$$X = \begin{pmatrix} \Delta X \\ \Delta X \\ c\delta_R(t) \end{pmatrix} \tag{2.14}$$

then,

$$X = A^{-1}l \tag{2.15}$$

The above algorithm was written as an m-file in Matlab, which is included in Appendix A. Once implemented, this m-file asks for an initial guess, it then calculates the estimated position based on the Newton-Raphson method.

2.5 Alternating Between Sensors in a Multi-sensor Systems

This paper is interested in the problem of multi-sensor systems where the position of a transmitter is calculated using multilateration (3 or more receivers) techniques. In such a system, there are often times that not all the sensors are available or the system requires receiver selection techniques to eliminate noisy results. Such selections methods consider the problem of choosing the best receivers to calculate a fit. This can result in the possibility of using a different constellation of sensors in each time interval. If the number of the sensors is increased or decreased, the estimated position of the transmitter can also change. This is illustrated in Figure 2.15 (b). In this figure, the position of the transmitter is estimated using four sensors as opposed to three that were previously used, shown in Figure 2.15 (a). When comparing Figure 2.15 (a) and Figure 2.15 (b) it can be seen that the estimated position of the transmitter has moved to a new location.

In a system in which this phenomenon happens constantly and at random, the measurement noise of the system can be different at any given instant of time. As stated earlier, in a multi-sensor system there are times that not all the sensors are available for calculating the estimate of a

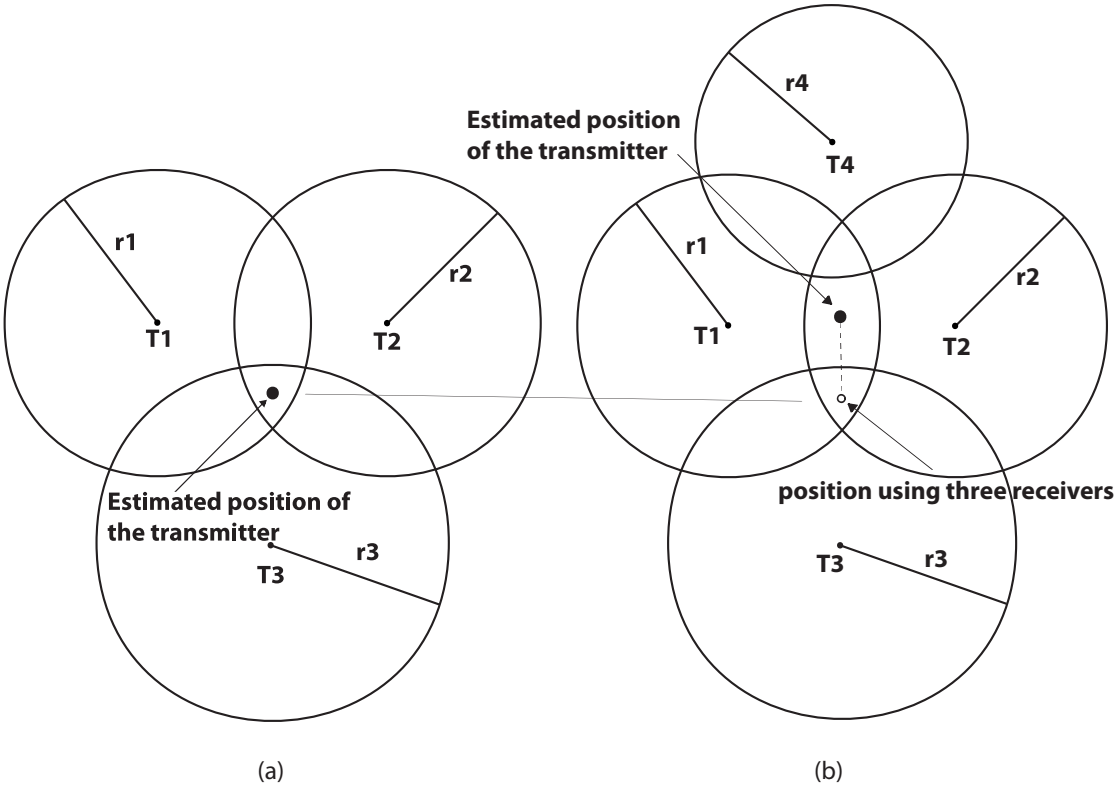


Figure 2.15 Multilateration when the number of sensors can change

parameter. This can be due to different factors such as the sensor not being able to sense any data, is out of reach or is simply turned off. This can change the measurement noise of the system. For a system that alternates between s different constellations of sensors, the measurement noises can be written as:

$$\mathcal{N}_t^s(\mu_s, \sigma_s) \tag{2.16}$$

where μ_s and σ_s^2 denote to the mean and the variance of the noise of each set.

As an example, one can think of the GPS system. In general, a GPS system selects the best set of satellites through selection algorithm to calculate the position of the receiver [21]. The noise model is different when 7 satellites are used as opposed to 6. In the case of a GPS system however, the impact on the performance is not significant due to the fact that the GPS unit has to travel a relatively long distance before the number of satellites changes. Moreover, the GPS system does not update as fast as an indoor positioning system and the required accuracy is not as high. In a

UWB indoor positioning system, the number of sensors used to calculate the position can change at a much faster rate than the GPS. This is due to the nature of the UWB system (employing short pulses, multipath problem, etc...) and the signals generated by a tag can be easily lost.

In a UWB positioning system the signal power intensity received by the sensors are not all equal to each other; some of the sensors are noisier than others. This is because of the presence of obstacles in the building, which the UWB can go through to get to a particular sensor, and also the presence of metallic objects, which can cause some additional noise. If the algorithm weighs the measurements from all the sensors equally, then it is risking the treatment of the noisy sensors in the same manner as the more reliable ones. The algorithms therefore usually exclude sensors that get a signal with power lower than a set threshold. For the receivers used in the calculation, the ones that get a stronger reading from a transmitter are weighed more into the equation than the receivers that have lower power readings. In general sensors that are closest to a tag are more likely to be selected for calculation [36].

The next section provides an example of a system where the sensor selection happens repeatedly over time and how this phenomenon can change the noise model of a system and therefore affect its performance.

2.6 System Modeling and Simulation

This section includes methods that were used to model a tracking system, and how such a system was simulated. The model created is based on the TOA technique. To better understand a positioning system, simulations are performed to analyze how the systems behaves under various conditions. All simulation programs were written in Matlab.

2.6.1 Solving Trilateration Using Least-Square Fit Method

As explained previously, solving the lateration problem is an essential part of a typical tracking system. The first step in simulating a positioning system is therefore solving the lateration problem mathematically. Earlier it was also shown that the position of a tag can be estimated once the sensed distance of the tag is known to at least three of the sensors (in 2D space). This section demonstrates how this position is estimated for a system with error. Figure 2.16 shows the trilateration setup for three sensors and a tag. In this figure, the estimated position of the tag is at (X, Y) and the estimated distance obtained from sensor A is the radius of the circle centered at A which is Ra.

This is the case for the other circles, as Rb and Rc (not shown in the figure) represent the estimated distance from the tag to the sensors B and C. One can see that the estimated position of the tag is then at a location which is closest to all the three circles. By definition, the shortest line from a point to a circle has to pass through the center of that circle. It can be concluded that the errors generated from each circle are the following:

$$E_a = R_a - \sqrt{(X - X_A)^2 + (Y - Y_A)^2} \tag{2.17}$$

$$E_b = R_b - \sqrt{(X - X_B)^2 + (Y - Y_B)^2} \tag{2.18}$$

$$E_c = R_c - \sqrt{(X - X_C)^2 + (Y - Y_C)^2} \tag{2.19}$$

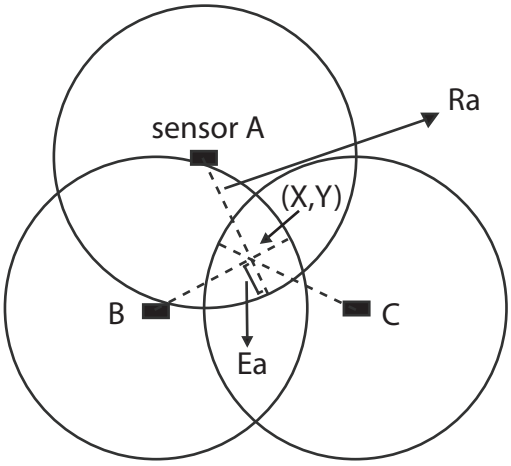


Figure 2.16 Trilateration setup for a system with error

To minimize these errors, the least square fit method is used. Therefore we are looking to minimize the following function:

$$f(x, y) = E_a^2 + E_b^2 + E_c^2 \tag{2.20}$$

Since each of the errors are nonlinear, the method used has to work with such functions. Such functions can be solved using Gauss-Newton or Quasi-Newton type methods as described in [36]. The Matlab function that deals with finding the least square of such nonlinear systems is *lsqnonlin()*. The return of this function is the estimated position of (X, Y) and the value of *f(x,y)*, which is the error. The user has the option of specifying how small the error should be and *lsqnonlin()* is terminated once the optimization parameter is satisfied [25].

2.6.2 Simulation

With the lateration problem solved, a tracking system can then be simulated in Matlab. The purpose of this simulation is to study the effect of sensor-selection in a multi-sensor system as discussed earlier. This simulation consists of estimating the position of a static tag using the lateration technique by employing three or four sensors. Figure 2.17 shows the simulation using four sensors. In this figure the centers of the circles are the actual positions of the sensors. Each circles is drawn using the values *Ra, Rb, Rc* and *Rd*, which are the estimated distances from each of the four sensors to the tag. The estimated position of the tag calculated is marked by 'x' on the figure.

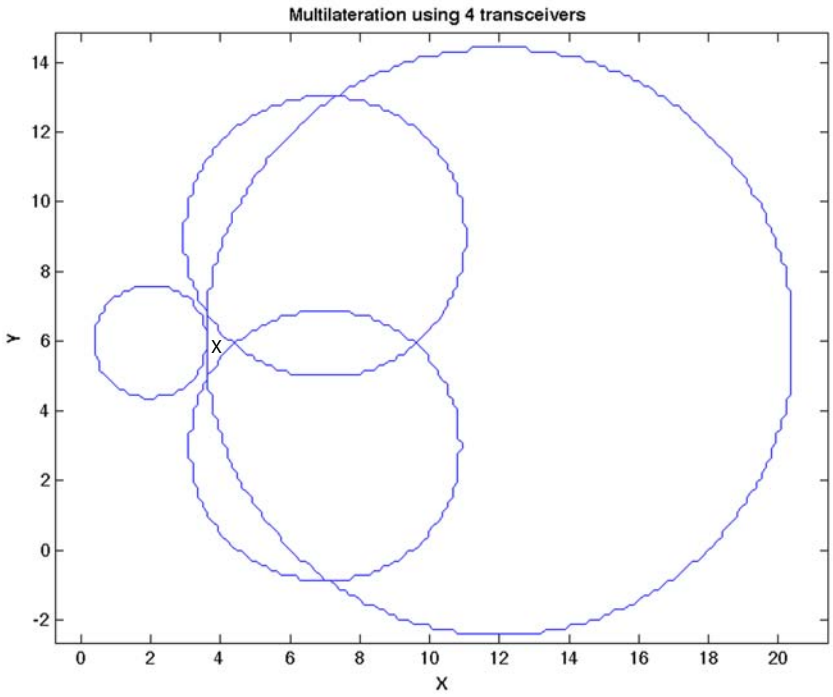


Figure 2.17 Multilateration simulation in Matlab using 4 sensors

Next, a system of three sensors is simulated. The simulation is performed for 2000 time steps.

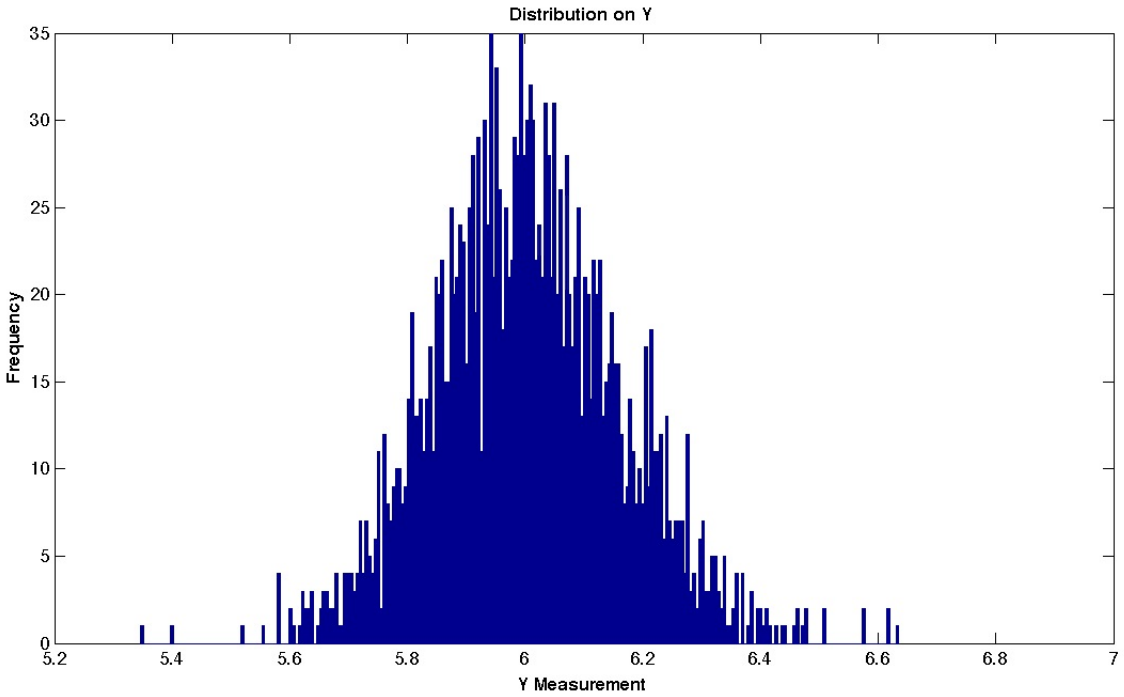


Figure 2.18 The distribution of a system employing three sensors on the y axis

The Matlab code written finds the estimated position of the tag by minimizing the least square function described earlier. For each time step, an x and a y position is calculated and stored. Figure 2.18 shows the distribution of the estimated y position of the tag, which fits a Gaussian model due to the Gaussian noises added. The distribution of the position in the y direction using four sensors is also Gaussian and similar to Figure 2.18 and therefore not shown.

The next step is to simulate a system that alternates between using three and four sensors at each consecutive time step. This means that there are two noise models, $\mathcal{N}^1$, the noise model that represents when the system uses 3 sensors, and $\mathcal{N}^2$ for when the system uses four sensors. Based on equation 2.16 we have,

$$\mathcal{N}_t^1(\mu_1, \sigma_1) \tag{2.21}$$

$$\mathcal{N}_t^2(\mu_2, \sigma_2) \tag{2.22}$$

Figure 2.19 shows the distribution of this system. It should be noted that this distribution is not

Gaussian even though it might appear to be one. The distribution of this system is the combination of two distinct distributions which were the results of using different numbers of circles to calculate the position. To see this better, the two distributions can be plotted on top of each other. Figure 2.20 shows the two distinct distributions that were resulted by running the alternating simulation. It can be seen that the two distributions have slightly different means and deviations. This is due to the fact that using three and four sensors result in different position estimations, hence different noise distributions.

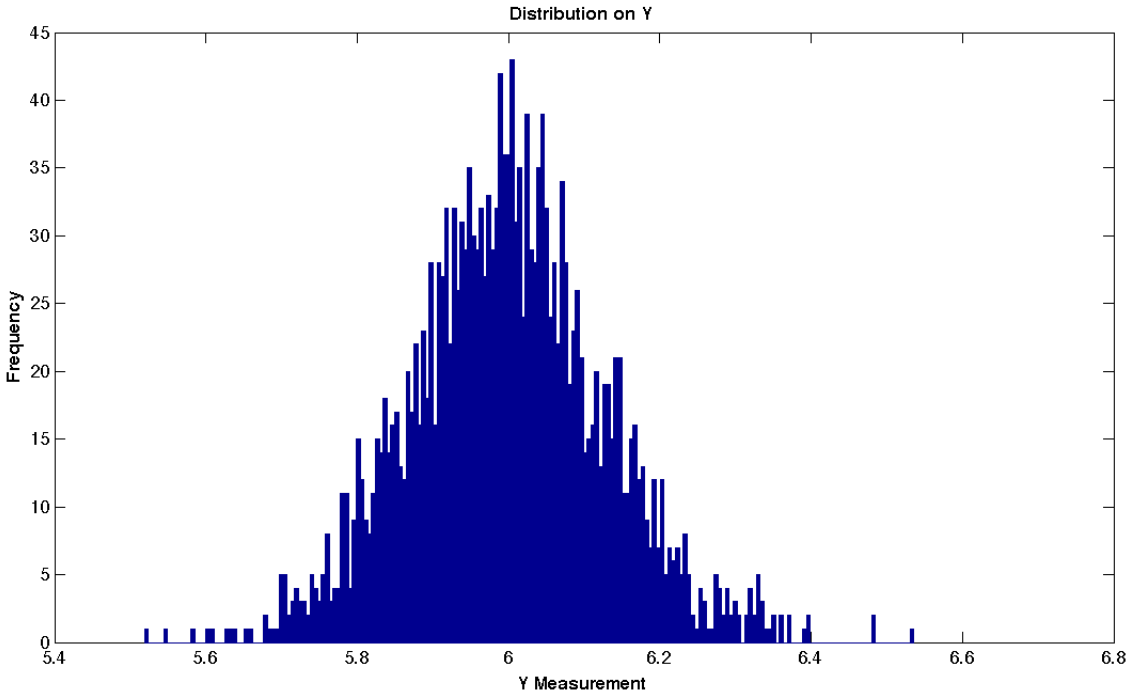


Figure 2.19 The distribution of a system alternating between 3 and 4 sensors on the y axis

The Gaussian noises added to each sensors were centered at zero (zero-mean) and had the same deviation value of 0.1 meters. A change in the mean of the noises would change the estimated position of the tag. This can happen in a real system, time delays caused by clock offset and delay in the arrival of the signal due to propagation through other mediums can be represented as a positive mean of the Gaussian noise added to the estimated distance of the tag to the sensor (radius of the circles created). Figure 2.21 demonstrates the results of changing the mean of one of the sensors (the fourth sensor) to 0.5 m.

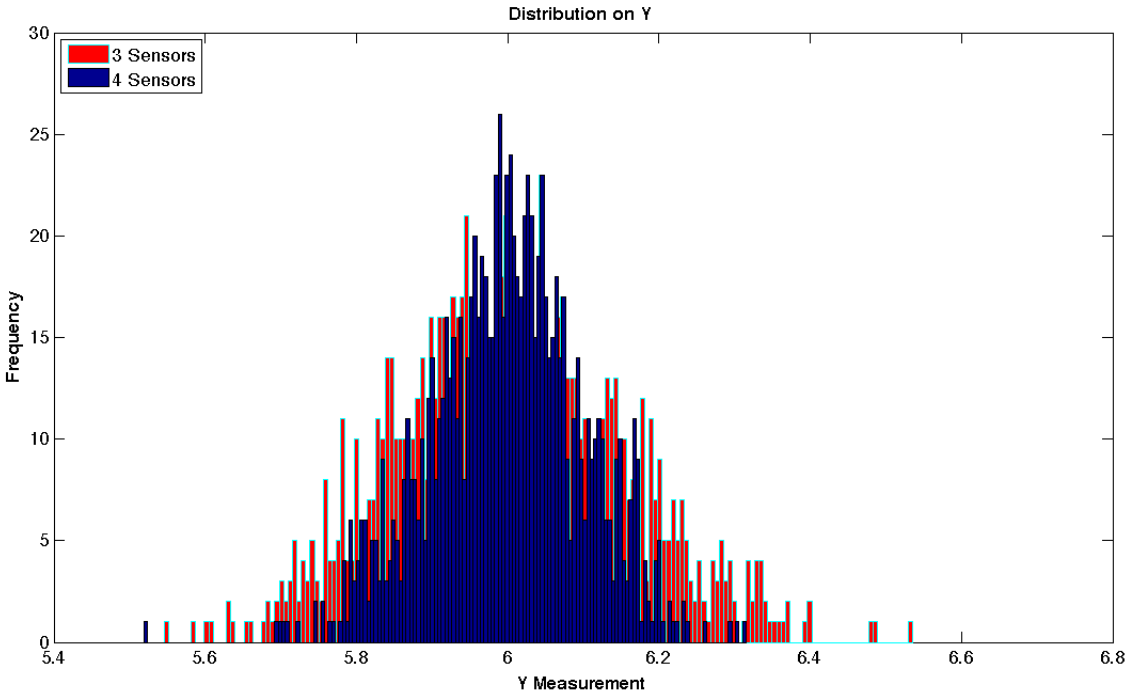


Figure 2.20 The two distinct y distributions of an alternating system

If we analyze the distribution in Figure 2.21 and break it down to two parts, we can better understand how the system is behaving. As shown before, when a system alternates between using two sets of sensors to calculate a parameter, it essentially has two distributions that can overlap. Also stated before, these two distributions have slightly different means, therefore they can not be immediately recognized on the distribution plot. However, if the means of these two distributions are visibly different, the 'bimodality' of the plot becomes apparent as shown in Figure 2.21. To further clarify this, Figure 2.22 demonstrates the two distinct distribution that have been obtained by the two different sets of sensors used. Once the two distribution in Figuree 2.22 are put together, Figure 2.21 is obtained.

2.7 Filtering

Filtering is the process of removing the unwanted part of a signal, namely the noise. The sources of noise in a positioning system can be a variety of things, from inaccurate system clocks to obstacles present between the tag and the sensors. There are various filtering algorithms available which are suitable for different systems. In selecting a filter, it is important to know the noise model of the

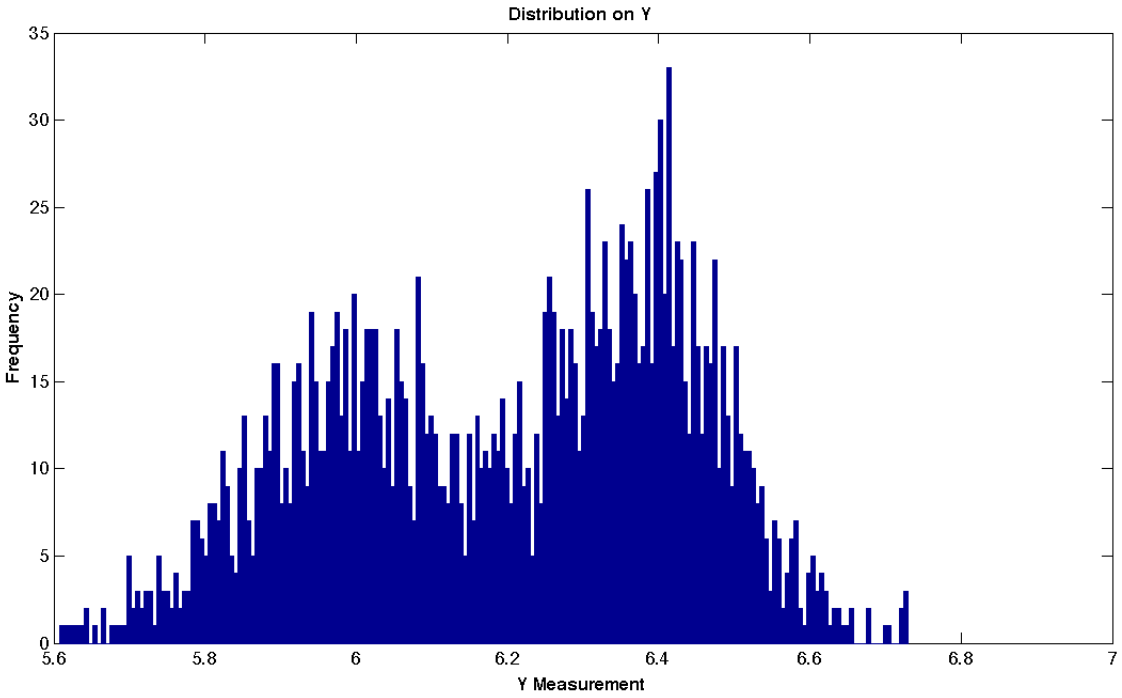


Figure 2.21 Bimodal distribution of a system that has a non-zero mean on one sensor

system as well as whether the system is linear or not.

To filter out the noise two widely used filters were considered, the Kalman filter and the particle filter. In the next two sections, the results of each filter will be discussed and the design of an optimized filter which addresses the issue of multi-modality will be given.

2.7.1 Kalman Filter

Kalman filter is one of the most widely used filters. In the software provided by the Ubisense system, the user is able to design a simple Kalman filter with parameters such as maximum and minimum speed limit and height of the tag. The Kalman filter designed for this project has four states, x and y, the positions on each axis, and their velocities $\dot{x}$ and $\dot{y}$. X_{t+1}, the state at time $t+1$, can be written in terms of the previous states at time t,

$$\overbrace{\begin{bmatrix} x_{t+1} \\ \dot{x_{t+1}} \\ y_{t+1} \\ \dot{y_{t+1}} \end{bmatrix}}^{X_{t+1}} = \overbrace{\begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{\Phi} \begin{bmatrix} x_t \\ \dot{x_t} \\ y_t \\ \dot{y_t} \end{bmatrix} + \overbrace{\begin{bmatrix} 0 \\ u_t \\ 0 \\ v_t \end{bmatrix}}^{U_t} \tag{2.23}$$

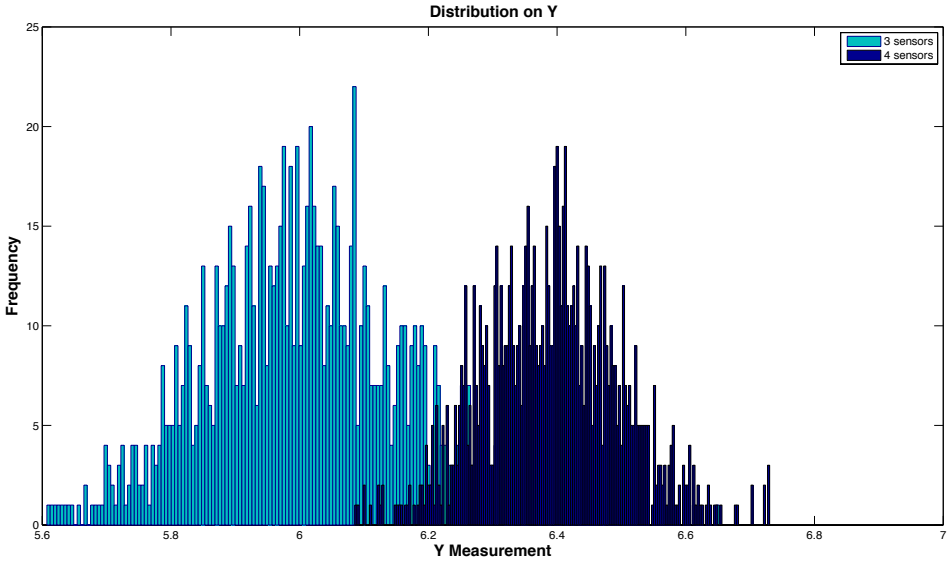


Figure 2.22 Two distinct modes of a system with a non-zero mean on one of its sensors

In the equation above, Φ is the state transition matrix and T is the sampling interval. The random dynamic noise, U_t, consists of u_t and v_t, the random accelerations on x and y during $t....t+1$. Similarly the measurement equation Y_t can be written in matrix form:

$$\overbrace{\begin{bmatrix} x_t \\ y_t \end{bmatrix}}^{Y_t} = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}^{M} \overbrace{\begin{bmatrix} x_t \\ \dot{x_t} \\ y_t \\ \dot{y_t} \end{bmatrix}}^{X_t} + \mathcal{N}_t(0, \sigma) \tag{2.24}$$

where M is the observation matrix and $\mathcal{N}_t(0, \sigma)$ is the measurement noise. Note that in the above equation, the measurement noises are zero-mean Gaussian.

With the model of the system created, the steps taken to execute the Kalman filter are the following:

1) Calculate the Kalman Gain matrix

$$K_t = S_{t,t-1}M'[MS_{t,t-1}M' + R]^{-1} \tag{2.25}$$

$$R = \begin{bmatrix} \sigma_{nx}^2 \\ \sigma_{ny}^2 \end{bmatrix} \tag{2.26}$$

where $S_{t,t-1}$ is the predictor covariance and R is the measurement noise covariance. σ_{nx}^2 and σ_{ny}^2

represent the measurement noise covariance on the x and y directions.

2) Update the state

$$X_{t,t} = X_{t,t-1} + K_t|Y_t - MX_{t,t-1}| \tag{2.27}$$

3) Update the predictor covariance

$$S_{t,t} = [I - K_t M]S_{t,t-1} \tag{2.28}$$

where I is the identity matrix

4) Predict the next state

$$X_{t+,t} = \Phi X_{t,t} \tag{2.29}$$

5) Estimate the next predictor covariance

$$S_{t+1,t} = \Phi S_{t,t}\Phi' + Q_t \tag{2.30}$$

6) Loop to step 1. t becomes $t + 1$

The predictor covariance is initialized to the dynamic covariance Q,

$$S_{0,0} = S_{1,0} = Q \tag{2.31}$$

where

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_u^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} \tag{2.32}$$

Note that since the velocity is assumed to be constant, the random changes on the position are zero. Therefore only the velocities have a dynamic noise covariance.

The designed Kalman filter is tested on a simulated system. For this simulation, a static object is placed at a known point and its position is calculated using the trilateration method on a multi-sensor system. As described earlier, this system alternates between a set of three and a set of four sensors. In Figure 2.23 the graph shows the estimated position of the stationary tag placed at unit 6 in the y direction. For the first 200 intervals, three sensors are used to estimate the position and

in the next 200 intervals four sensors are used. In other words, the system switches between noise models described in equation 2.16. The mean of the noise of the 3-sensor system is about zero as evident on the graph. It is however apparent that the mean of the noise of the 4-sensor system is not zero. In fact, the mean of the estimated position is about 6.4. The reason for this shift can be due to various factors explained before. Error in the position of a sensor, inaccuracies in the clock offset, and the delay that is added to the signal while the signal from a sensor travels through walls and objects, can all contribute to this shift of the center of the noise.

In equation 2.16 the s different sets of noise models that a multi-sensor system uses were shown. Each of these noise models can have different values for the mean and the covariance. In the Kalman filter designed, each of the elements in R matrix can be changed to a different measurement noise covariance when the system alternates between the sets of sensors. The R matrix however, does not capture the mean, μ, and assumes that it is zero. It is therefore apparent that the Kalman filter is only optimal for a zero-mean Gaussian system. Moreover, for the bimodal system, the Kalman filter is not an appropriate filter to use. The results of the Kalman filter is discussed in the next chapter.

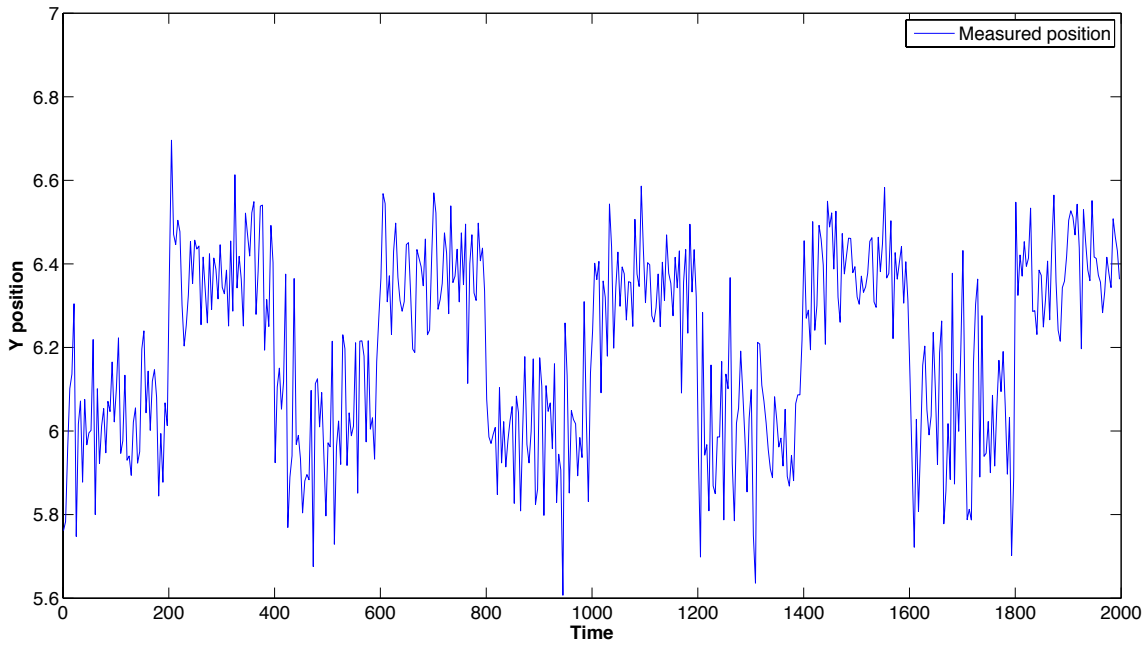


Figure 2.23 Estimated position of the stationary tag

2.7.2 Particle Filter

A particle filter was designed based on [9] and implemented in Matlab. The Matlab code is included in Appendix A. The particle filter designed uses the four states previously described in equation 2.23. Let $f()$ be a matrix of i state transition equations, where

$$X_{t+1} = f(X_t, \gamma_t) \tag{2.33}$$

where γ_t is the dynamic noise.

Let $g()$ be a matrix of j observation equations where

$$Y_t = g(X_t, \eta_t) \tag{2.34}$$

where η_t is the measurement noise.

Having defined the $f()$ and $g()$ the following steps are taken to implement the particle filter:

1) Chose a number of particle M

2) Initialize the particles. The states can be set to zero with equal weights in the absence of knowledge.

$$\chi = \{X^m, w^m\}_{m=1}^M = \{0, \frac{1}{M}\}_{m=1}^M \tag{2.35}$$

3) At each time step of the running filter, the following takes place:

a. Each particle's state is propagated through the state transition equation with a random dynamic noise

$$\{X_t^m\}_{m=1}^M = \{f(X_{t-1}^m), \gamma_t^m\}_{m=1}^M \tag{2.36}$$

b. The weights of the particles are updated according to:

$$w_t^m = w_{t-1}^m p(y_t | X_t^m) \tag{2.37}$$

c. Particles are normalized so their weights sum to 1.

$$\left\{ w^m = \frac{w^m}{\sum_{m=1}^{M} w^{(m)}} \right\}_{m=1}^M \tag{2.38}$$

d. The desired value is calculated, such as the expected value.

$$E[x] = \sum_{m=1}^{M} X^m \cdot w^{(m)} \tag{2.39}$$

The ideal measurement for each particle is Z_t^m calculated using the observation equations. It can be written as:

$$Z_t^m = g(X_t^m, 0) \tag{2.40}$$

And $p(y_t|x_t)$ can be calculated for each particle using the particle's state and the measurement noise distribution. For example, if the measurement noise is Gaussian, modeled as $\mathcal{N}_t(0, \sigma_{meas})$, then

$$p(Y_t|X_t^m) = \frac{1}{\sqrt{(2\pi\sigma_{meas}^2)}} e^{\frac{Y_t - Z_t^{m\,2}}{2\sigma_{meas}^2}} \tag{2.41}$$

When the system alternates between different sets of sensors, the distribution changes as shown previously. In the particle filter designed for a multi-sensor system, there needs to be a condition that determines which probability distribution should be used and assign the weights of the particles accordingly. In the case of the simulated system presented, the system alternates between two sets of sensors, therefore there are two different noise models. To address this, the weights of the particles are updated by $p_s(Y_t|X_t^m)$, where s denotes to different noise models the system switches between as described in equation 2.16 . Similarly to before we have:

$$Z_t^m = g(X_t^m, \mathcal{N}(\mu_s, 0)) \tag{2.42}$$

$$p_s(Y_t|X_t^m) = \frac{1}{\sqrt{(2\pi\sigma_s^2)}} e^{\frac{Y_t - Z_t^{m\,2}}{2\sigma_s^2}} \tag{2.43}$$

To update the particles weights, the filter decides which noise model is used and then selects the appropriate $p_s(Y_t|X_t^m)$ for each particle. The change in the mean and the variance of each noise model, μ_s and σ_s , can be accounted for in Z_t^m and $p_s(Y_t|X_t^m)$ respectively, whereas the Kalman filter assumes all the noise distributions are zero-mean Gaussian. This is the advantage of the designed particle filter over the Kalman filter. The result of the particle filter is demonstrated in the next chapter.

CHAPTER 3

RESULTS

This chapter discusses the experimental and simulation results.

3.1 Accuracy of Ubisense system

As mentioned before, the accuracy of a point on a single axis in the system is calculated by subtracting the measured value minus the ground-truth value. This was done for all the points taken throughout the Shoothouse. The xy-plane accuracy was then computed and the results are shown in Figure 3.1. The x and y axis represent the coordinates of a point and the intensity of the color of each grid represents the accuracy (darker areas have higher accuracies). Similarly, Figure 3.2 shows the 3D plot of the accuracy, the values of the z-axis here represent the accuracies.

It can be seen that some areas have higher accuracies than others. The accuracies of rooms 1 and room 4 (refer to map of the Shoothouse in Figure 2.1) are lower than Room 3, 5 and 6. This however can be explained by the properties of UWB explained in Chapter 1. Room 3, 5 and 6 get better LOS coverage than other areas of the Shoothouse, this is the primary factor behind relatively higher accuracies in these areas. Also more sensors point to these areas and therefore there is relatively better coverage in Room 3, 5 and 6 than in other parts of the Shoothouse. After the experiment was performed, it was noted that the main sensor covering Room 4 had inaccurate coordinates input during the calibration process (about 10 cm off in the x direction). Since the readings done in Room 4 put more weight on this sensors, the accuracies in this room are lower. This sensor is also used as one of the main sensors in Room 1 which also explains the low accuracy in that room. One other factor that might effect the accuracy is the presence of metallic objects in the hallway area immediately next to Room 4. This hallway is blocked as it can be seen in Figure 2.1; walls and materials which were not used in the building of the Shoothouse were stacked in this hallway. Since some of the signals from neighboring sensors have to propagate through this dense area, delays to the signals are added which increase the error in the position estimation as explained in Chapter 1.

The histogram of the accuracy is shown in Figure 3.3. It can be seen that most of the accuracy in the Shoothouse is under 50 cm. The overall mean of the accuracy is calculated to be 55.6 cm and the standard deviation is 48.6 cm.

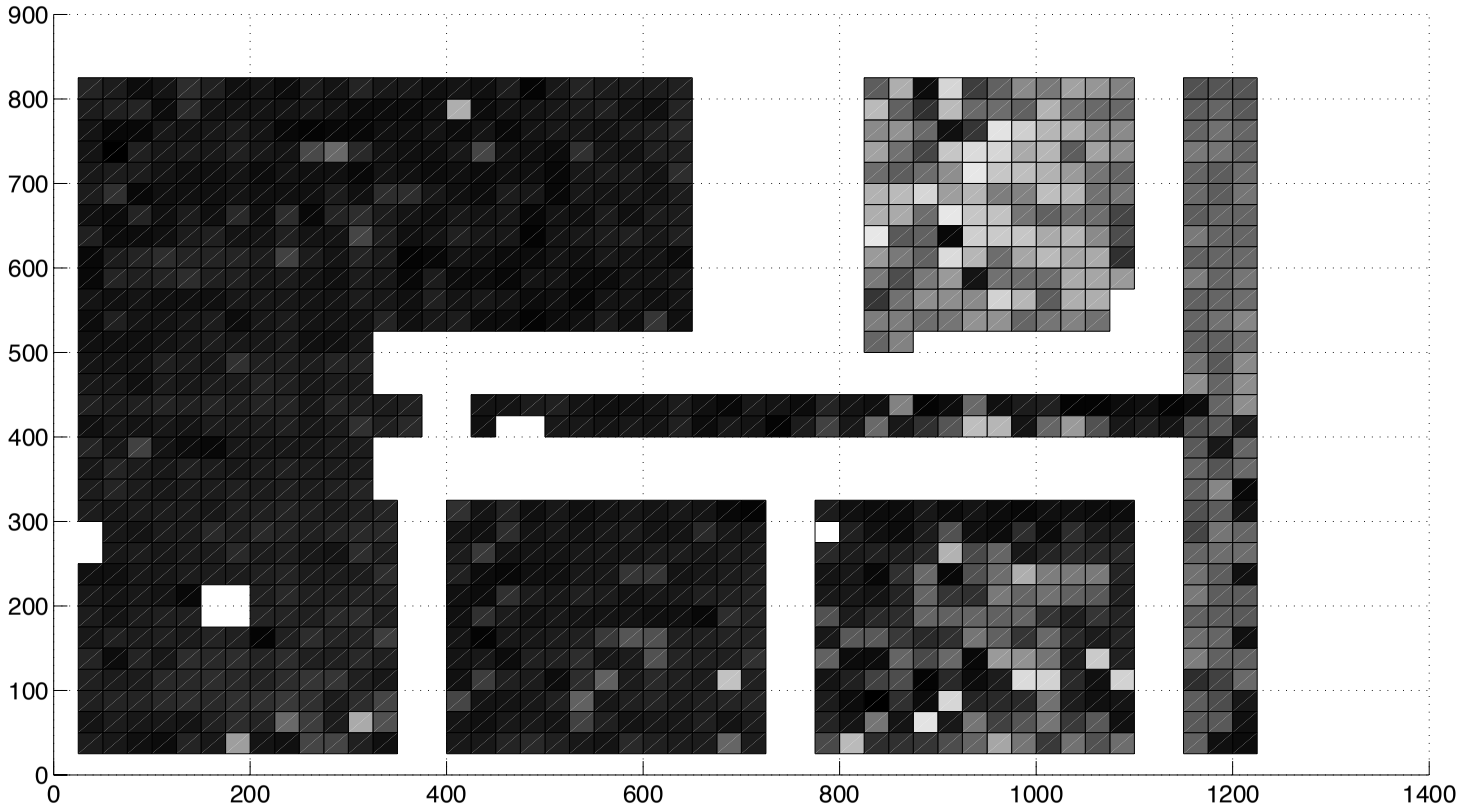


Figure 3.1 2D plot of the accuracy of the Ubisense system. The grayscale represents range of 10 (dark)-200 (light) *cm* accuracy

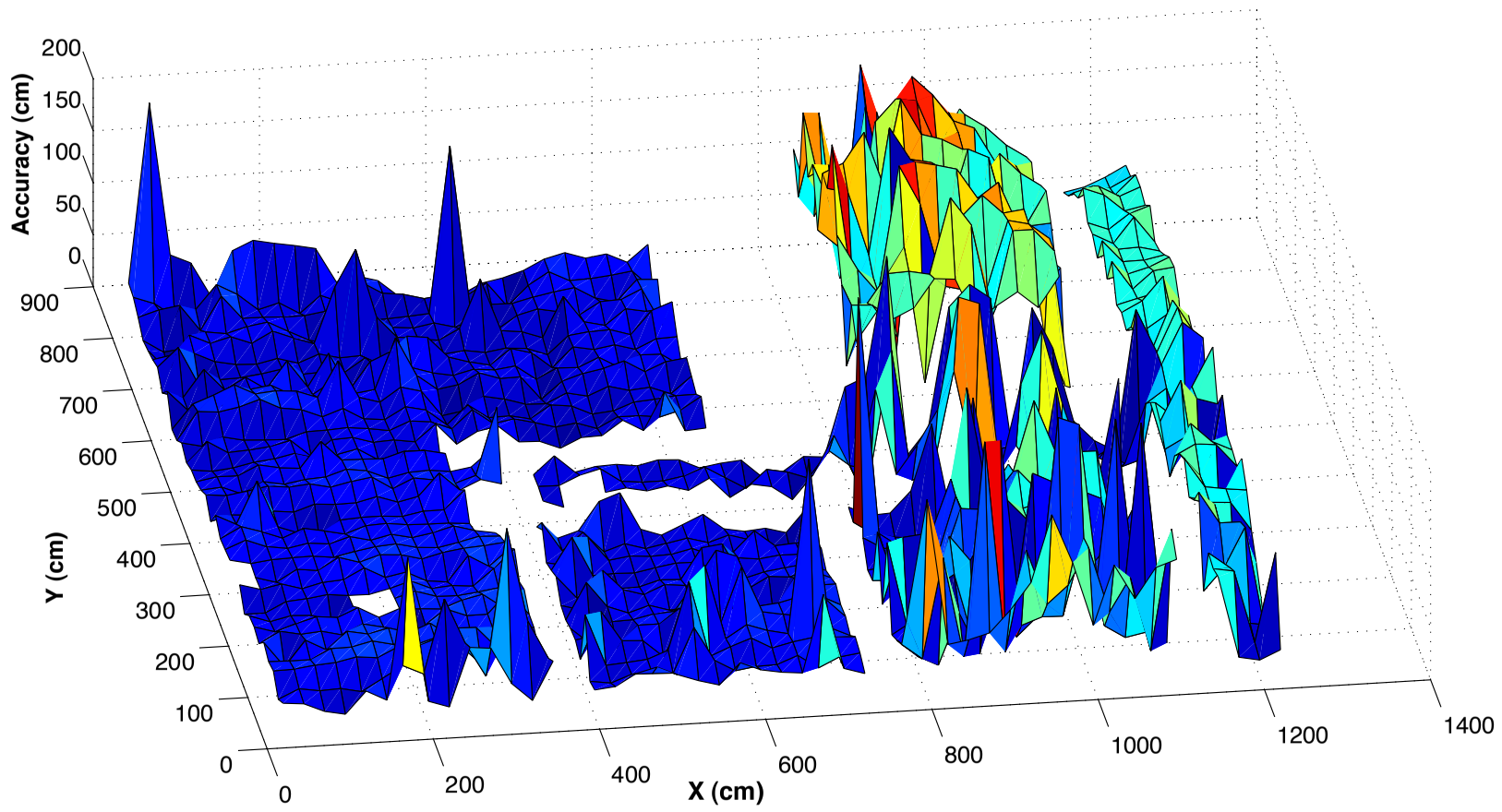


Figure 3.2 3D plot of the accuracy of the Ubisense system

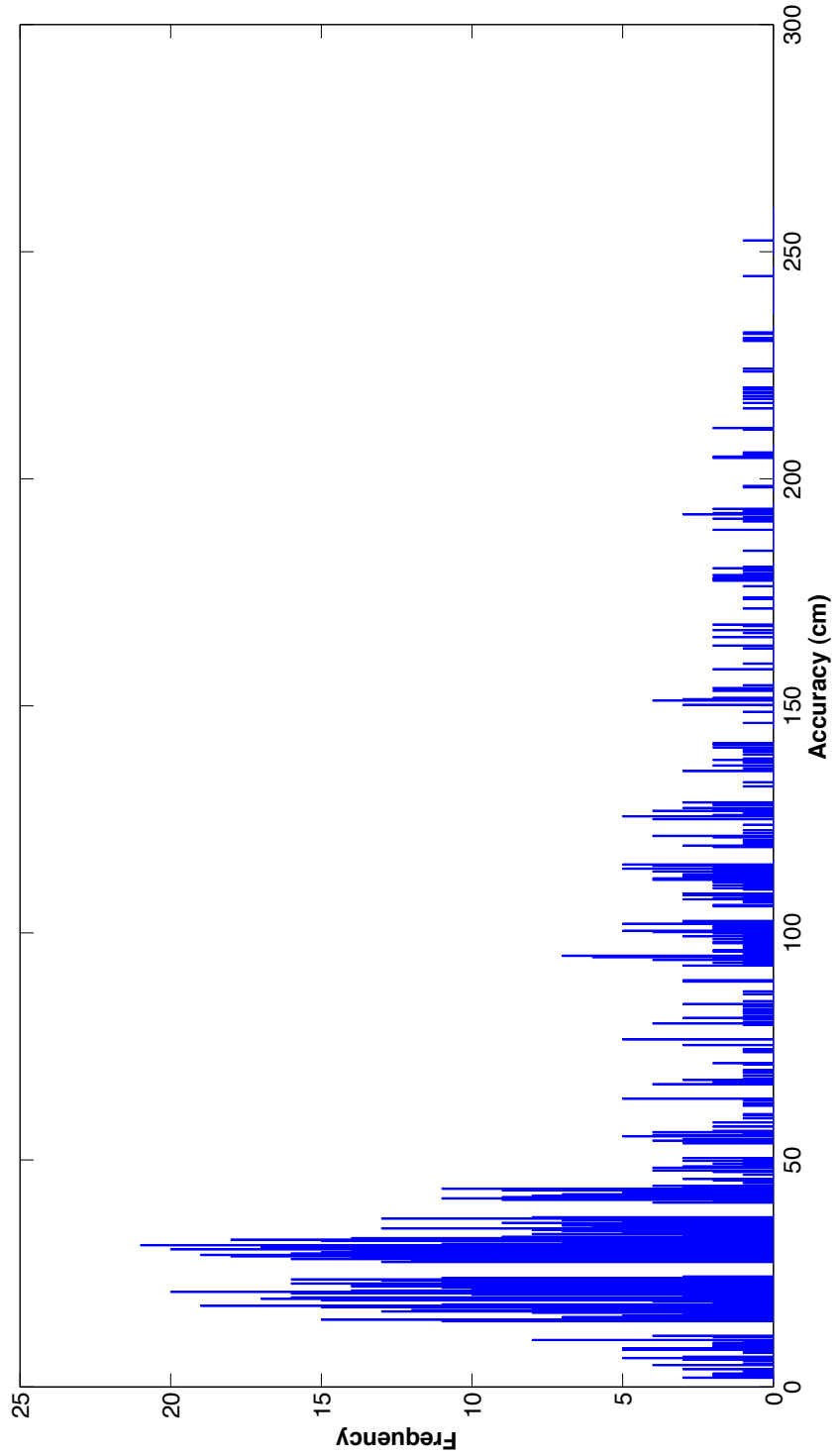
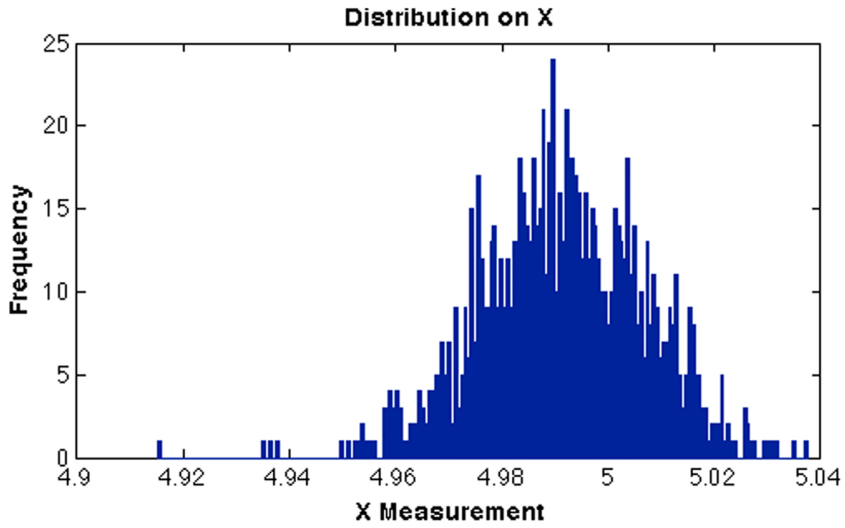


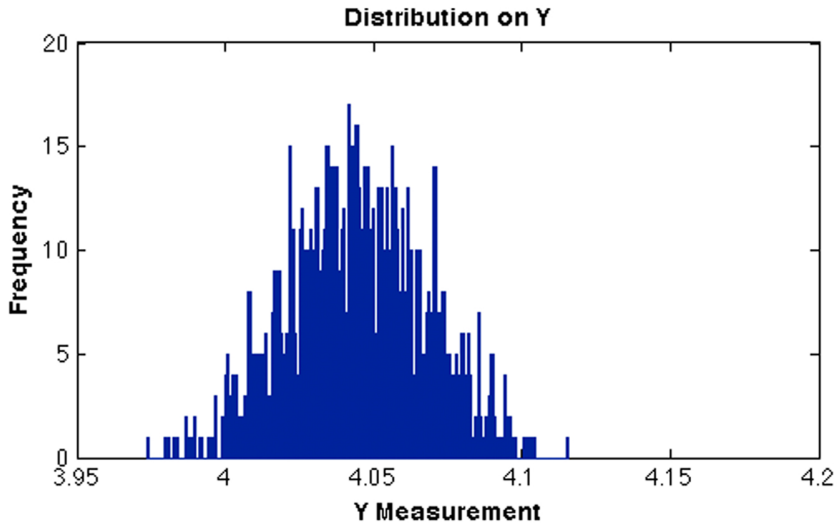
Figure 3.3 Histogram of the accuracy of the Ubisense system

3.2 Muti-modality of Ubisense system

This section demonstrates the multi-modality of the Ubisense system. In the Ubisense system, the user can see which sensor is used at each time interval on the graphical interface. If a sensor is used, a green line is drawn from the sensor towards the tag. Although this can be seen on the screen, the results cannot be saved while taking the measurements using C-code. In other words, we do not know which sensor is used for a given measurement. Therefore a series of tests are performed to show how the system performs when different sensor sets are used.

After the calibration of the system, all the sensors are turned off except two that have a good coverage of an area of interest. A total of 1000 sample points are taken by placing a tag on a tripod and measuring its location using these two sensors. The histogram results are shown in Figure 3.4. It can be seen that the histograms represents Gaussian distributions. Next a third sensor is turned on. Now the system can alternate between the three sensors. If the calibration process is accurate, the peaks of the distributions lie on top of each other, which is the case for the x-axis in Figure 3.5. On the y-axis however, the histogram clearly demonstrates the multi-modality of the system. The same is true when four sensors are used as shown in Figure 3.6. When all the sensors are used in taking measurements, some of the histograms shown in Figure 3.7 look like single modal distributions. However in reality they are multi-modal distributions, where the modes lie on top of each other as stated before. Since it is known that the algorithm puts more weight on some of the sensors, altering the calibration parameters of these sensors would cause the system inaccuracy to go up and in theory the multi-modality would appear on the histogram. Two sensors that had LOS conditions were selected and their calibration parameters were slightly changed. Then similar to before, 1000 measurements were taken and the histograms were plotted. The different modes in the distribution can now be seen since the modes do not lie on top of each other (Figure 3.8). By turning these two sensors off, the inaccurate sensors are taken off when calculating the location of a tag and the modes lie on top of each other again as shown in Figure 3.9.





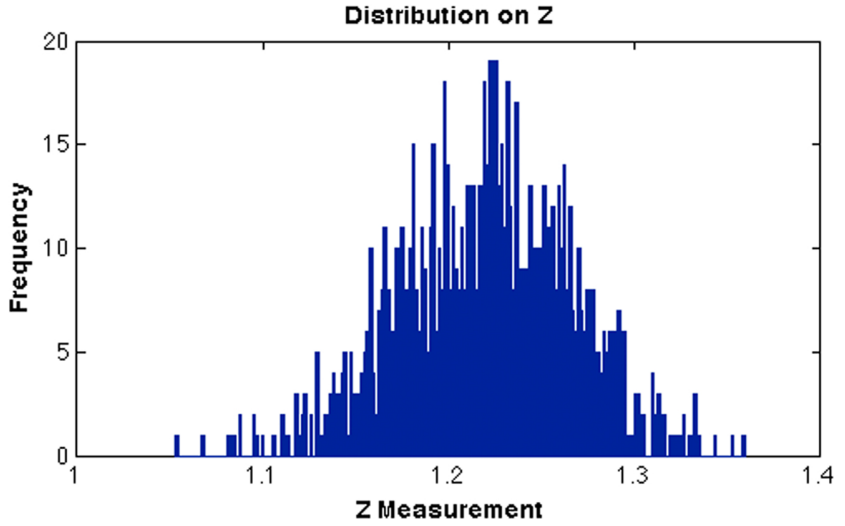
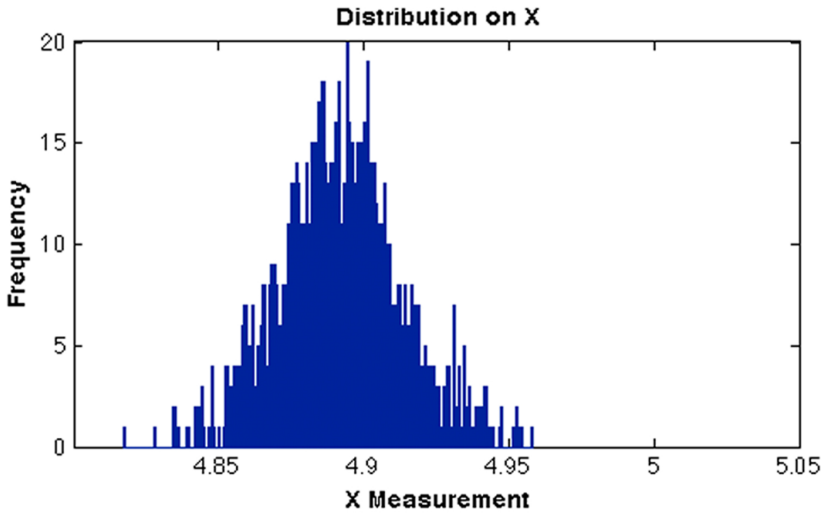
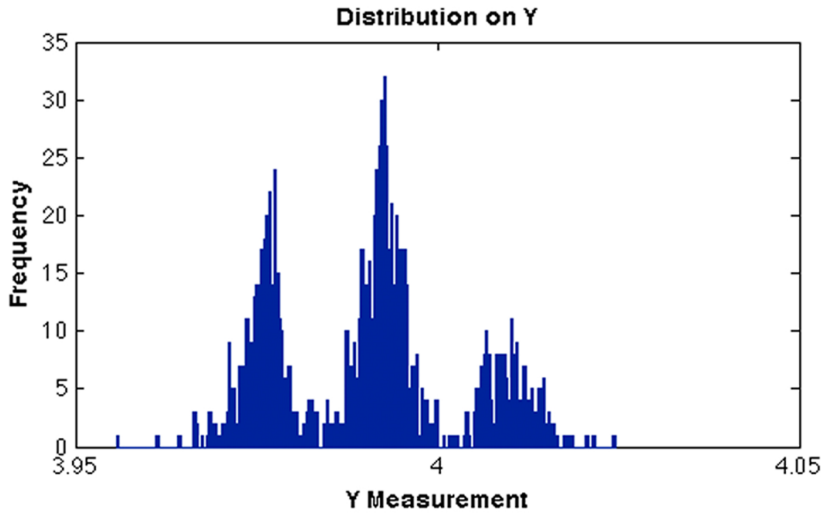


Figure 3.4 Measurements taken using 2 sensors after calibration





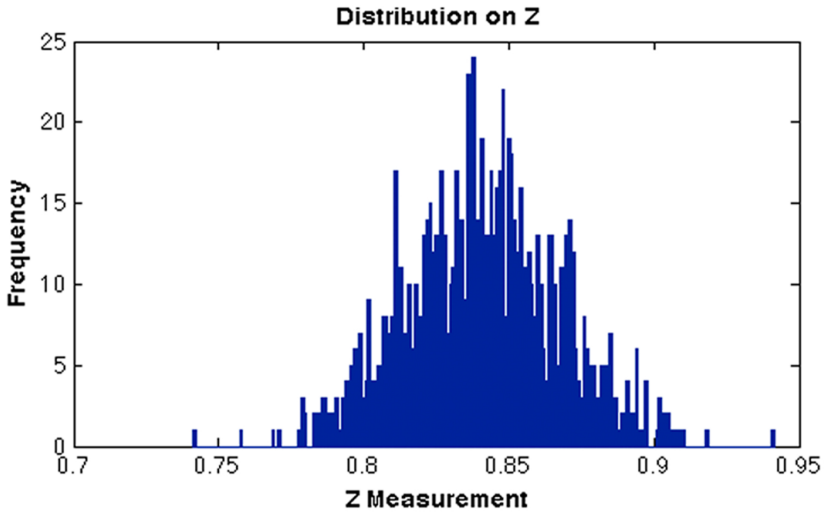


Figure 3.5 Measurements taken from 3 of the sensors after calibration

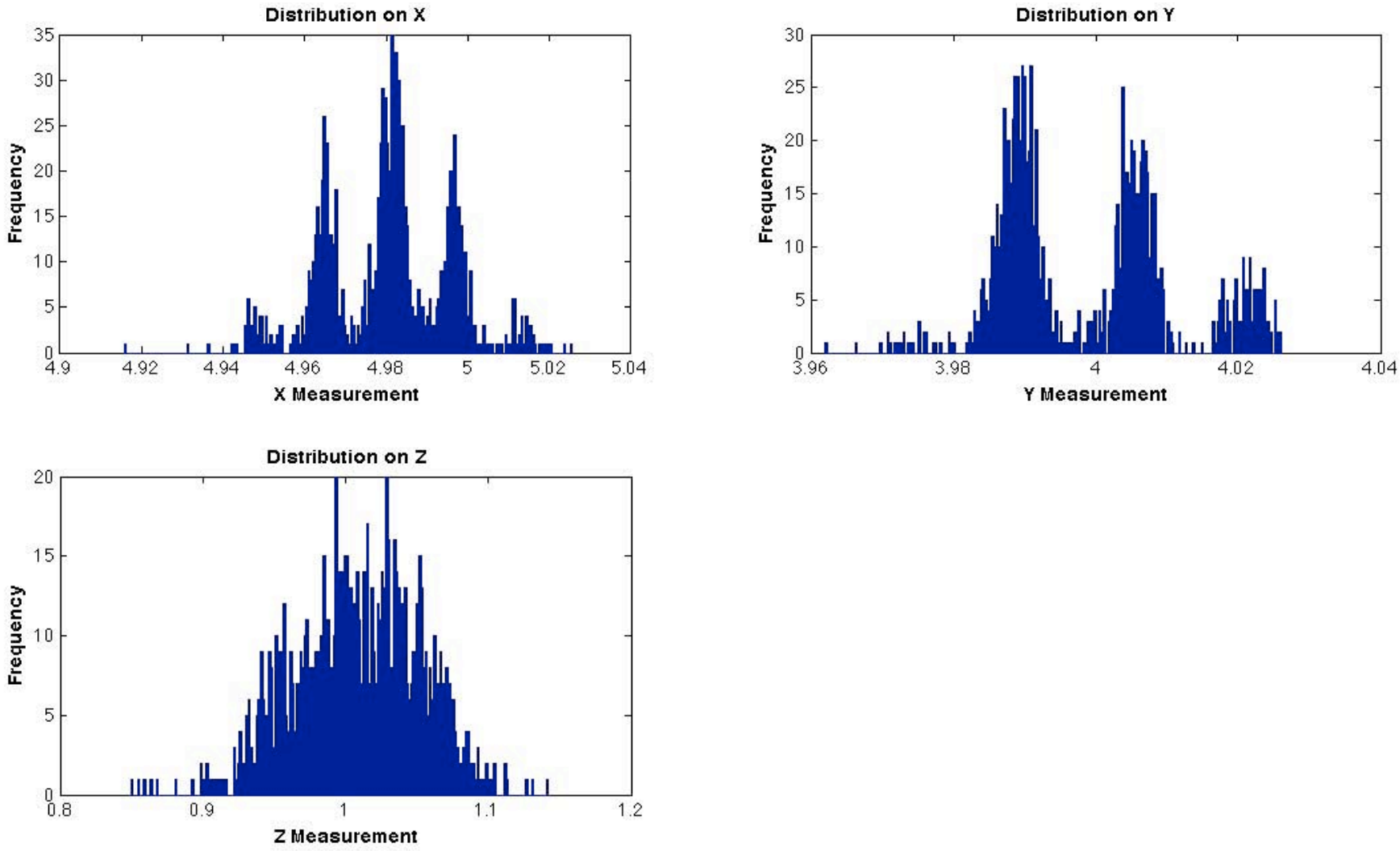


Figure 3.6 Measurements taken from 4 of the sensors after calibration

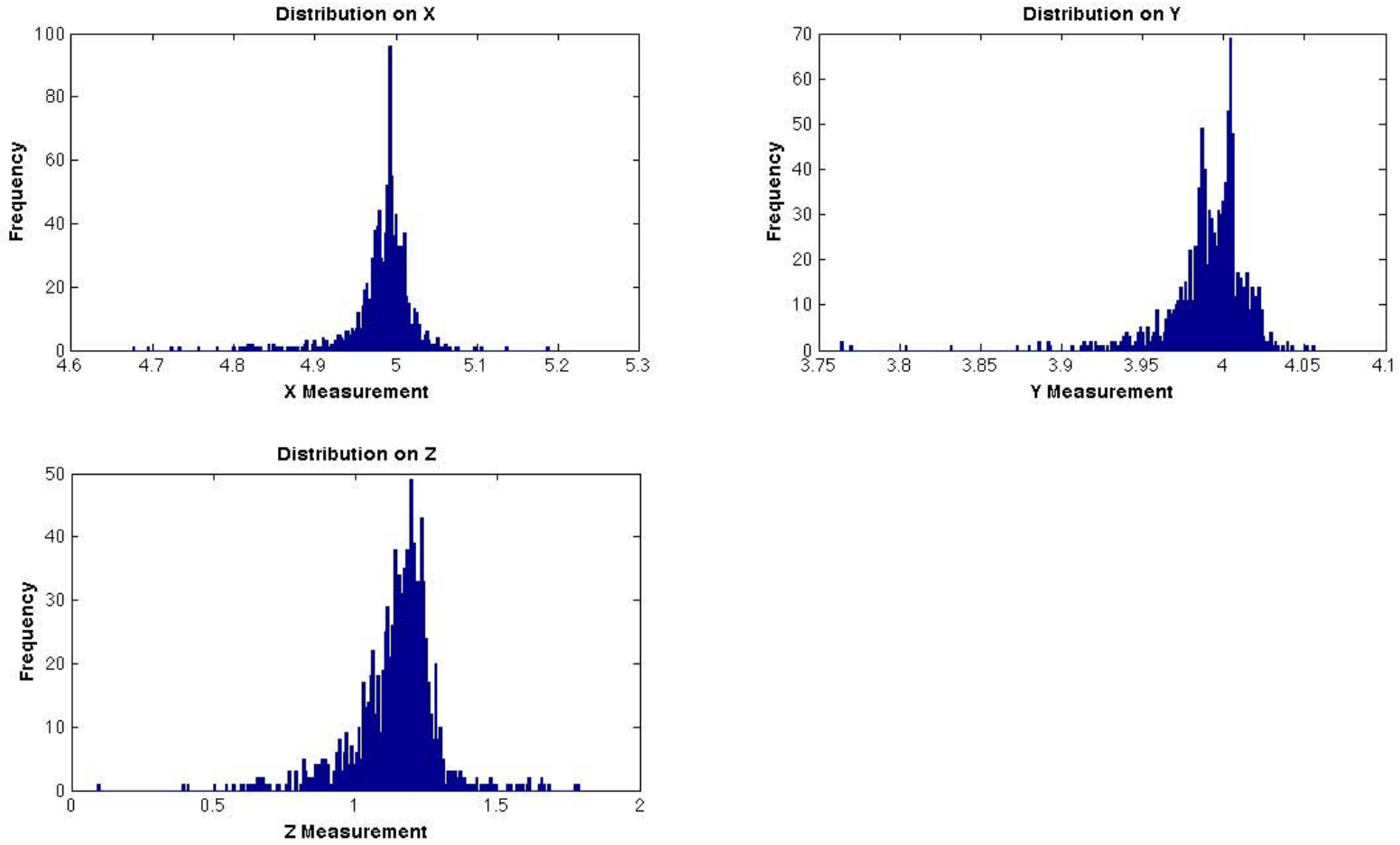


Figure 3.7 Measurements taken from all the sensors after calibration

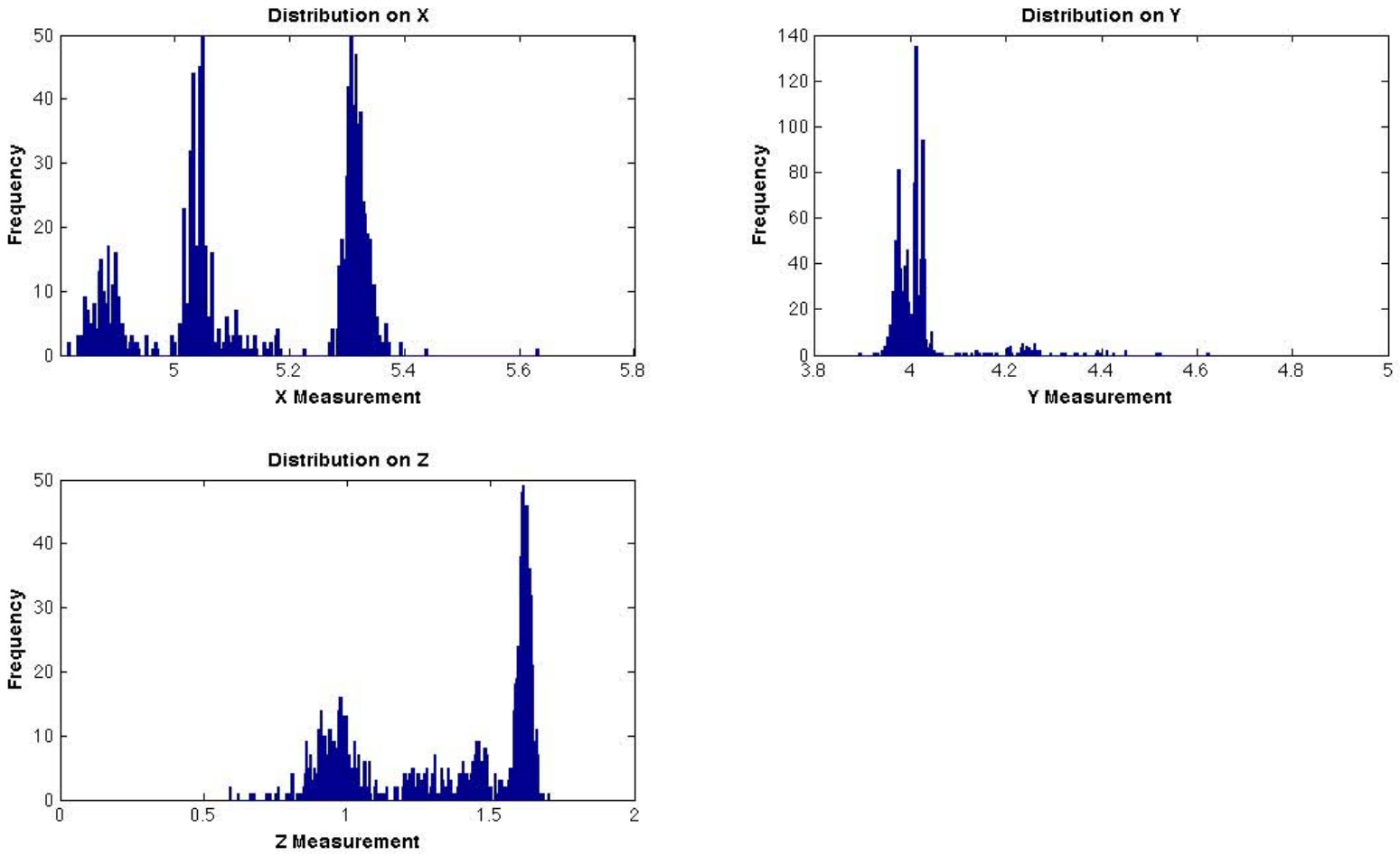


Figure 3.8 Measurements taken from all the sensors after calibration while main two sensors are altered

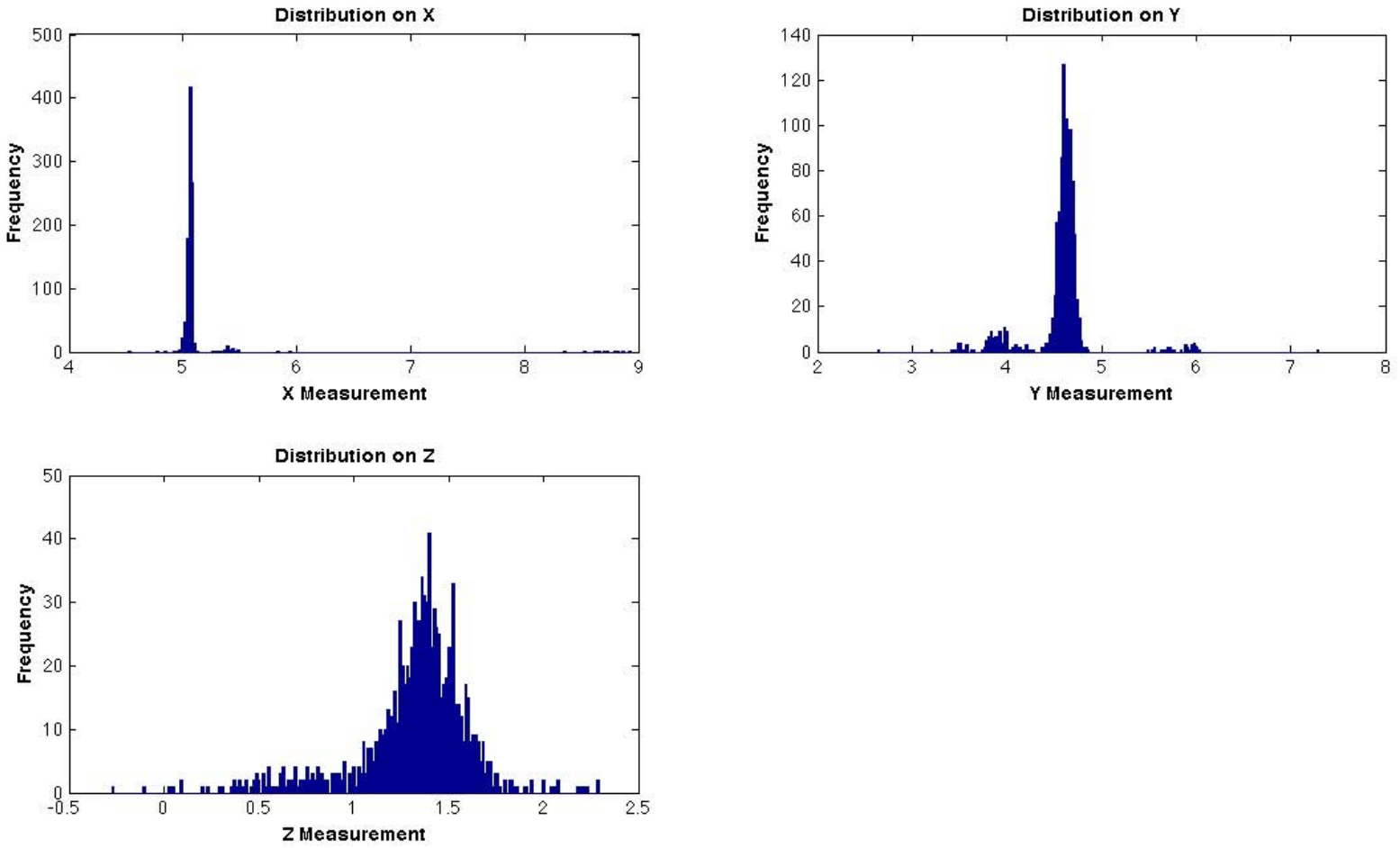


Figure 3.9 Measurements taken from all the sensors after calibration while main two sensors are altered

3.3 Results of the Simulation and Filters

This section discusses the results of the simulations and the two filters that were used.

3.3.1 Kalman Filter

As explained in Chapter 1, the Kalman filter works with noise models that are zero-mean Gaussian. In the alternating system simulated in the previous chapter, the Kalman filter is not expected to be optimal. The Kalman filter is used to track the measurements created by the simulated system explained in the previous chapter (section 2.6.1); these measurements are also shown in Figure 3.10. In this simulation the object is stationary and positioned at 6 in the y-direction. The estimated results of the Kalman filter are shown on this graph which clearly do not give the desired results of tracking the stationary tag. As a result a filter that addresses the issue of multi-modality and the shift in the mean of the noise should be designed and implemented when using a multi-lateration tracking system.

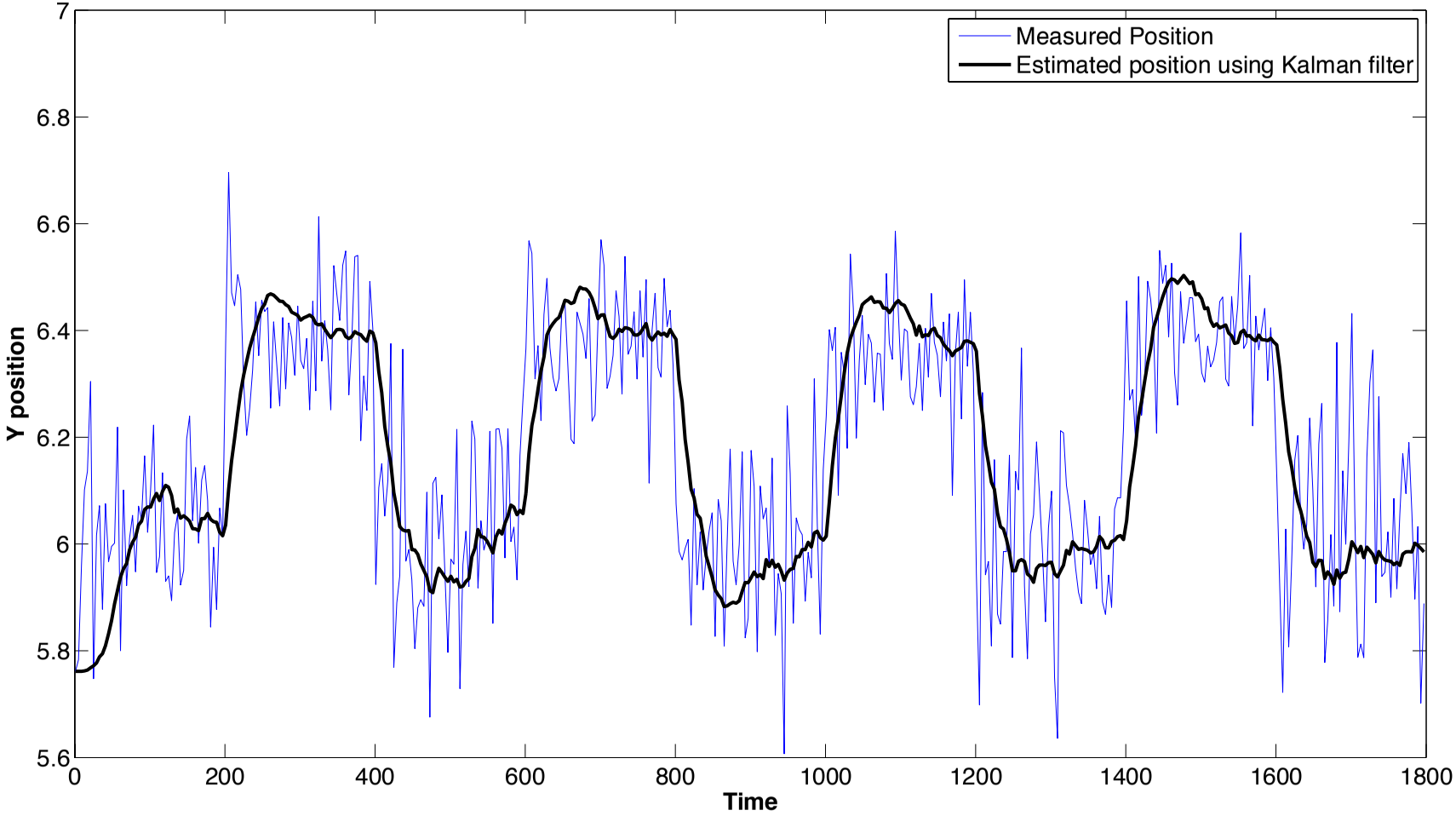


Figure 3.10 Results of Kalman Filter

3.3.2 Particle Filter

This section contains the result of the particle filter for the simulated system described in Chapter 3. For the designed particle filter to be optimal, the filter needs to know when the system alternates between the set of sensors and which noise model should be used. Since the sensor selection for the Ubisense tracking system could not be recorded, this filter was not implemented on the real data.

Figure 3.11 shows the result of the designed particle filter with 1000 particles as described in section 2.6.2 implemented on the simulated system. In this simulation the object is positioned at 6 on the y-direction. The graph shows to the position of the object in the y-axis at each time step. The filtered estimate of the position at each time step is also shown. It can be seen that the filter is able to estimate the position of the object even when the system alternates between two sets of sensors.

The filter can also be applied to a moving object created by the simulation. The solid line in Figure 3.12 represents the actual position of the moving tag on the y-axis. It can be seen that the particle filter is able to filter some of the noise and stay closer to the true position of the tag.

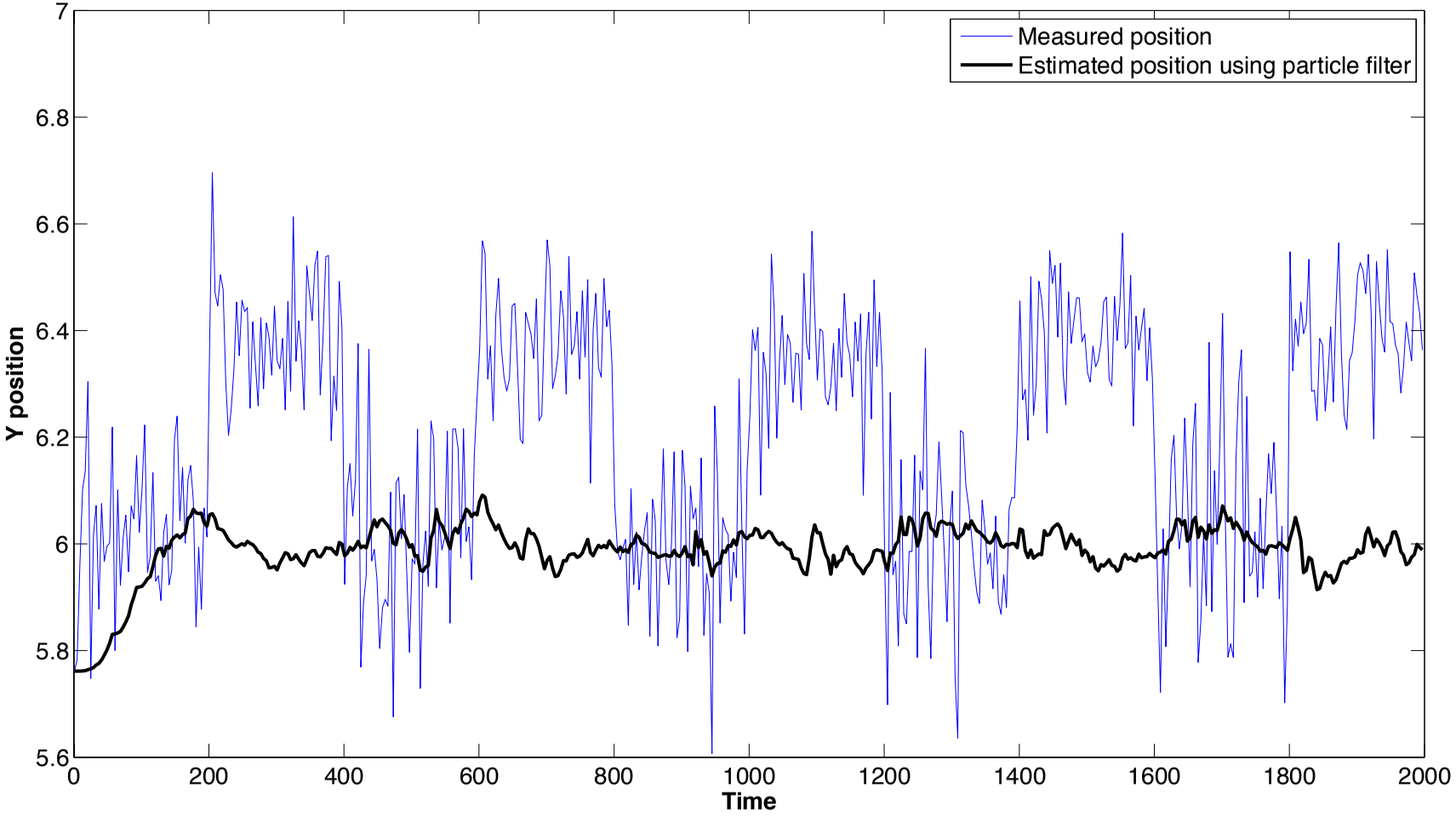


Figure 3.11 Particle filter results

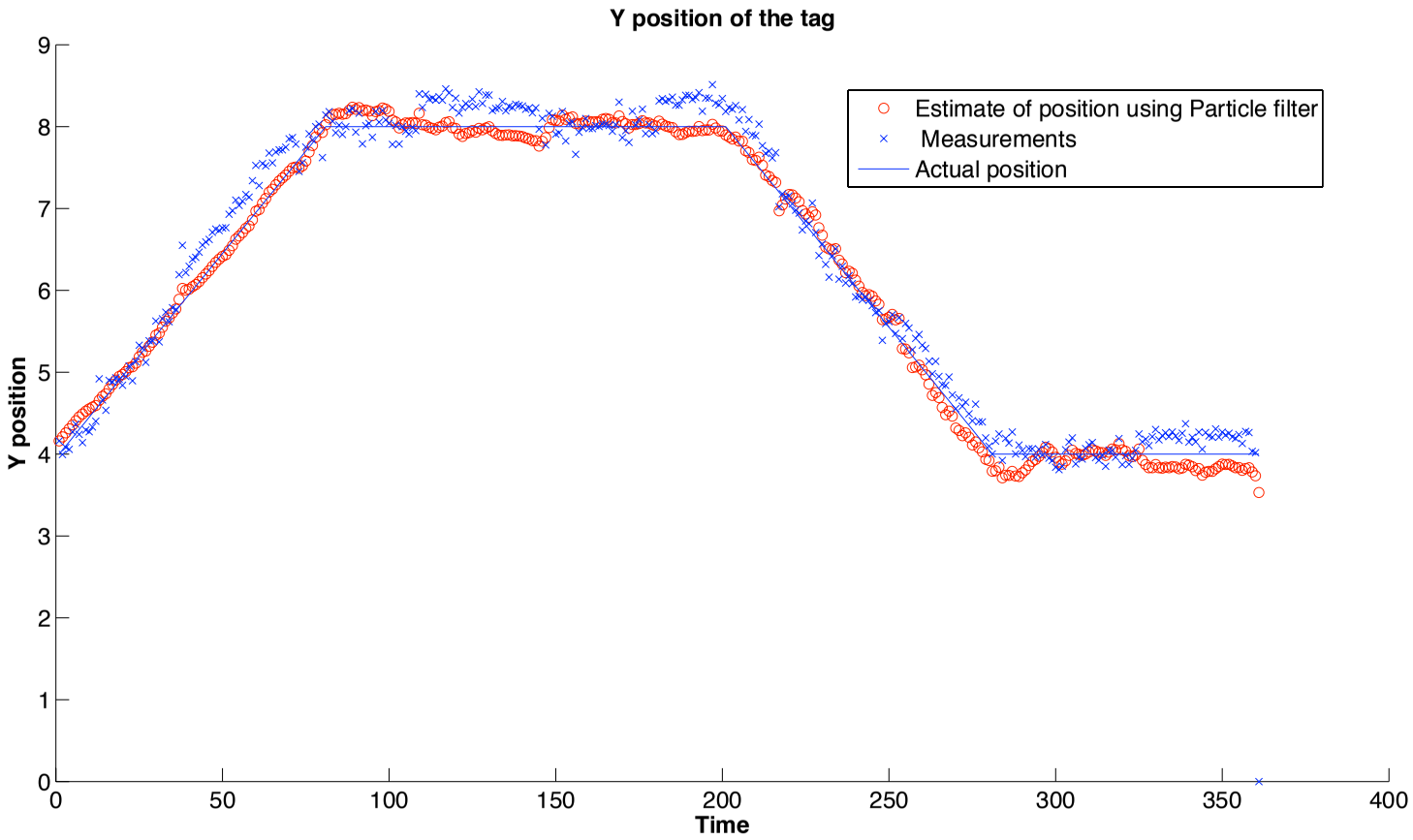


Figure 3.12 Results of Particle filter on a moving object

CHAPTER 4

CONCLUSION

In this paper, the accuracy of the Ubisense system tracking system was evaluated in a multi-room building. By taking measurements throughout the entire building, it was shown that the accuracy is not uniform and in fact is different in every room. Based on the properties of the UWB, the factors that effected the accuracy were then explained. Next, the multi-modality of the Ubisense system was demonstrated, which explains why a Kalman filter would not be optimal for such a tracking system.

In the next part of the paper, a multi-sensor tracking system that alternates between sensors was simulated. This system was shown to also be multi-modal. The performance of the Kalman filter was then evaluated on such a system, and it was shown that this filter does not effectively reduce the noise and produce reliable estimates. It was explained that this is due to the fact that the Kalman filter is only optimized when the noise model is represented by a zero-mean Gaussian random distribution. A system that alternates between sensors, requires a filter which addresses the issue of multi-modality. A multi-modal particle filter was designed and tested on the simulated system. It was shown that the performance of the designed particle filter is superior when compared to the conventional Kalman filter.

4.0.1 Future Work

The future work includes testing the accuracy of the tracking system on moving objects , and addressing the inaccuracy problem in certain spots of the Shoothouse and the ways it can be improved. Changing the sensor configuration and adding more sensors in highly noisy areas are some of the measures that can be taken to improve the overall performance of a UWB tracking system in such a building. It was shown that the filter designed was effective in reducing the noise in the simulated system. To truly evaluate its performance however, the filter needs to be tested on a real system. Unlike the Ubisense system, such a tracking system would need to let the user record which sensor is being used for a given measurement. The filter can then use the appropriate noise model to produce better estimates of the position.

APPENDIX

APPENDIX A

The following code is written to solve the trilateration problem using Newton-Raphson method.

```
% This is an example of solving trilateration using Newton-Raphson method.
function newtonraphsontrilateration;

% The following are the position of the center of each sensor (satellite or
% tranceiver)
XA=[5,2];
XB=[7,4];
XC=[9,2];

% The distance from the receiver to each transceiver (tag to sensor)
DA=2.3;
DB=2.1;
DC=2.5;

%drawing the circles
fh1 = @(x,y) sqrt((x-XA(1,1)).^2 + (y-XA(1,2)).^2 - DA^2);
fh2 = @(x,y) sqrt((x-XB(1,1)).^2 + (y-XB(1,2)).^2 - DB^2);
fh3 = @(x,y) sqrt((x-XC(1,1)).^2 + (y-XC(1,2)).^2 - DC^2);

%plotting the circles
ezplot(fh1,[1,13],[-2,8]);
hold;
ezplot(fh2,[1,13],[-2,8]);
ezplot(fh3,[1,13],[-2,8]);
axis square;

%asking the user to input the initial guess for x,y
x0=input('please input the initial guess for x     ');
y0=input('please input the initial guess for y      ');


%writing out the Newton-Raphson equations
p1 = @(x,y) sqrt((x-XA(1,1))^2 + (y-XA(1,2))^2);
p2 = @(x,y) sqrt((x-XB(1,1))^2 + (y-XB(1,2))^2 );
p3 = @(x,y) sqrt((x-XC(1,1))^2 + (y-XC(1,2))^2);

dp1dx=@(x,y) -(XA(1,1)-x)/sqrt((x-XA(1,1)).^2 + (y-XA(1,2)).^2);
dp1dy=@(x,y) -(XA(1,2)-y)/sqrt((x-XA(1,1)).^2 + (y-XA(1,2)).^2);

dp2dx=@(x,y) -(XB(1,1)-x)/sqrt((x-XB(1,1)).^2 + (y-XB(1,2)).^2);
dp2dy=@(x,y) -(XB(1,2)-y)/sqrt((x-XB(1,1)).^2 + (y-XB(1,2)).^2);

dp3dx=@(x,y) -(XC(1,1)-x)/sqrt((x-XC(1,1)).^2 + (y-XC(1,2)).^2);
dp3dy=@(x,y) -(XC(1,2)-y)/sqrt((x-XC(1,1)).^2 + (y-XC(1,2)).^2);
```

```
for i=0:1:5;

A=[dp1dx(x0,y0),dp1dy(x0,y0),-1;
       dp2dx(x0,y0),dp2dy(x0,y0),-1;
       dp3dx(x0,y0),dp3dy(x0,y0),-1];

l=[DA-p1(x0,y0);DB-p2(x0,y0);DC-p3(x0,y0)];
d=A^-1*l;

x0=x0+d(1,1)
y0=y0+d(2,1)
end
scatter(x0,y0,'red','x','LineWidth',2)
```

The following is a Kalman filter written as an m-file in Matlab

```
%this filter is designed to filter out the noise for an alternating
%multisensor positioning system. Xest3 & Xest4 are obtained from simulation
%m-file called multisimulation.m

function Kalmanfiltering
%map=imread('map.ppm');%loading the map of the building
%double=im2double(map);%converting to double for graphing purposes
clear
Xtag=[4,6];
%actual position of the tag
load Xestimated
%this load Xest3 & Xest4 which are
%the estimated positions of the tag using 3&4 sensors
Xestboth=zeros([1,2,2000]);
S=200;
%alternating at every S steps

Rmatrix=zeros(1,2000);
%look in the loop below for the description of Rmatrix
%Xestboth alternates between the two systems at every values of S
for i=1:2*S:(2000)
% X
Xestboth(1,1,i:1:i+S-1)=Xest3(1,1,i:1:i+S-1);
Xestboth(1,1,i+S:1:i+2*S-1)=Xest4(1,1,i+S:1:i+2*S-1);
% Y
Xestboth(1,2,i:1:i+S-1)=Xest3(1,2,i:1:i+S-1);
Xestboth(1,2,i+S:1:i+2*S-1)=Xest4(1,2,i+S:1:i+2*S-1);
% Rmatrix is used later on in the main loop to determine which variance is
% used, variance for a 3-sensor system or 4-sensor.
Rmatrix(1,i:1:i+S-1)=3;
Rmatrix(1,i+S:1:i+2*S-1)=4;

end

T=1;
%ax=.25;
%ay=.25;
```

```
%B=600;

%Distribution of Random Dynamic Noise
sigmax=.0002;%
sigmay=.0002;%

%Dynamic Model Covar
Q=[0,0,0,0;0,(sigmax)^2,0,0;0,0,0,0;0,0,0,(sigmax)^2];
%sigma dist. of random dynamic noise
Sstate=Q;

%State Transition Matrix
phi=[1,T,0,0;0,1,0,0;0,0,1,T;0,0,0,1];

%Observation Matrix
M=[1,0,0,0;0,0,1,0];

% measurment noise covariance
%for 3 sensor system
sigmanoisex3=std(Xest3(1,1,1:2:end))^2;
sigmanoisey3=std(Xest3(1,2,1:2:end))^2;
%for 4-sensor system
sigmanoisex4=std(Xest4(1,1,1:2:end))^2;
sigmanoisey4=std(Xest4(1,2,1:2:end))^2;
%for a alternating system
sigmanoisexboth=std(Xestboth(1,1,1:2:end))^2;
sigmanoiseyboth=std(Xestboth(1,2,1:2:end))^2;

R3=[sigmanoisex3,0;0,sigmanoisey3];
R4=[sigmanoisex4,0;0,sigmanoisey4];
Rboth=[sigmanoisexboth,0;0,sigmanoiseyboth];
%R=[.25,0;0,.25];

%XY=load('XY.txt'); % the file with the x,y coordinates
X(1,:)=Xestboth(1,1,:); % x coordinate
Y(1,:)=Xestboth(1,2,:); % y coordinate
size1=size(Y);
duration=size1(1,2);     %how long the loop will run for
Spredict=phi*Sstate*phi'+Q;
Measurement=[X(:,:);Y(:,:)];
Xstate(1:4,1)=[Measurement(1,1);0;Measurement(2,1);0];
%starting at the actual position to speed up the filter
Xpredict(1:4,1)=[Measurement(1,1);0;Measurement(2,1);0];

%kalman filter using a single
%noise model (3&4 combined, Xestboth)
R=Rboth;
%R=[50,0;0,50];
for t=2:1:duration;
    Sstateminus1=Spredict;
    K(:,:)=Sstateminus1*(M')*(M*Sstateminus1*M'+R)^(-1);
    %update state:
    Xstate(1:4,t)=Xpredict(1:4,t-1)+K(:,:)*(Measurement(1:2,t)
        -M*Xpredict(1:4,t-1));
```

```
    %update predictor covariance
    Sstate=(eye(4)-K(:,:)*M)*Sstateminus1;
    %predict
    Xpredict(1:4,t)=phi*Xstate(1:4,t);
    Spredict=phi*Sstate*phi'+Q;
end



% now using alternating noise models/variances in the loop.
Sstate=Q;
Spredict=phi*Sstate*phi'+Q;
%same as before
Measurement=[X(:,:);Y(:,:)];
%same as before
Xstateboth(1:4,1)=[Measurement(1,1);0;Measurement(2,1);0];
% starting at the actual position to speed up the filter
Xpredictboth(1:4,1)=[Measurement(1,1);0;Measurement(2,1);0];

for t=2:1:duration;
    %the value of R is determined by Rmatrix, which says if 3 or 4 sensors
    %were used at the time.
    if Rmatrix(1,t)==3;
        R=R3;
    else
        R=R4;
    end

    Sstateminus1=Spredict;
    K(:,:)=Sstateminus1*(M')*(M*Sstateminus1*M'+R)^(-1);
    %update state:
    Xstateboth(1:4,t)=Xpredictboth(1:4,t-1)+K(:,:)*(Measurement(1:2,t)
        -M*Xpredictboth(1:4,t-1));
    %update predictor covariance
    Sstateboth=(eye(4)-K(:,:)*M)*Sstateminus1;
    %predict
    Xpredictboth(1:4,t)=phi*Xstateboth(1:4,t);
    Spredict=phi*Sstateboth*phi'+Q;
end

save('Kalmanfiltering.mat');
scatter([1:1800],Xstate(3,1:1800),'green');
%filtered position by using a bimodal noise model
hold
scatter([1:1800],Xstateboth(3,1:1800),'r');
%filtered position by alternating between the two noise models
scatter([1:1800],Xestboth(1,2,1:1800),'b');
% the actual measurement recored by the sensors of the tag

set(gca,'FontSize',14)      %sets the font size of the plot to 14
xlabel('Time','FontSize',14,'FontWeight','b');
ylabel('Y position','FontSize',14,'FontWeight','b');
title('Y position of the tag','FontWeight','b','FontSize',14);
set( gcf, 'Color', [1,1,1] );
```

```
legend('bimodal noise model','alternating btwn 2 noise models',
'actual measurements',1);
```

The following is a particle filter written as an m-file in Matlab

```
% This is an example of solving trilateration using Newton-Raphson method.
function newtonraphsontrilateration;

% The following are the position of the center of each sensor (satellite or
% tranceiver)
XA=[5,2];
XB=[7,4];
XC=[9,2];

% The distance from the receiver to each transceiver (tag to sensor)
DA=2.3;
DB=2.1;
DC=2.5;

%drawing the circles
fh1 = @(x,y) sqrt((x-XA(1,1)).^2 + (y-XA(1,2)).^2 - DA^2);
fh2 = @(x,y) sqrt((x-XB(1,1)).^2 + (y-XB(1,2)).^2 - DB^2);
fh3 = @(x,y) sqrt((x-XC(1,1)).^2 + (y-XC(1,2)).^2 - DC^2);

%plotting the circles
ezplot(fh1,[1,13],[-2,8]);
hold;
ezplot(fh2,[1,13],[-2,8]);
ezplot(fh3,[1,13],[-2,8]);
axis square;

%asking the user to input the initial guess for x,y
x0=input('please input the initial guess for x     ');
y0=input('please input the initial guess for y      ');


%writing out the Newton-Raphson equations
p1 = @(x,y) sqrt((x-XA(1,1))^2 + (y-XA(1,2))^2);
p2 = @(x,y) sqrt((x-XB(1,1))^2 + (y-XB(1,2))^2 );
p3 = @(x,y) sqrt((x-XC(1,1))^2 + (y-XC(1,2))^2);

dp1dx=@(x,y) -(XA(1,1)-x)/sqrt((x-XA(1,1)).^2 + (y-XA(1,2)).^2);
dp1dy=@(x,y) -(XA(1,2)-y)/sqrt((x-XA(1,1)).^2 + (y-XA(1,2)).^2);

dp2dx=@(x,y) -(XB(1,1)-x)/sqrt((x-XB(1,1)).^2 + (y-XB(1,2)).^2);
dp2dy=@(x,y) -(XB(1,2)-y)/sqrt((x-XB(1,1)).^2 + (y-XB(1,2)).^2);

dp3dx=@(x,y) -(XC(1,1)-x)/sqrt((x-XC(1,1)).^2 + (y-XC(1,2)).^2);
dp3dy=@(x,y) -(XC(1,2)-y)/sqrt((x-XC(1,1)).^2 + (y-XC(1,2)).^2);

for i=0:1:5;

A=[dp1dx(x0,y0),dp1dy(x0,y0),-1;
```

```
        dp2dx(x0,y0),dp2dy(x0,y0),-1;
        dp3dx(x0,y0),dp3dy(x0,y0),-1];

l=[DA-p1(x0,y0);DB-p2(x0,y0);DC-p3(x0,y0)];
d=A^-1*l;

x0=x0+d(1,1)
y0=y0+d(2,1)
end
scatter(x0,y0,'red','x','LineWidth',2)
```

REFERENCES

[1] H. Arslan, Z. Chen, and M. Di Benedetto. *Ultra Wideband Wireless Communication*. John Wiley Sons Inc., first edition, 2006.

[2] E. Brookner. *Tracking and Kalman Filtering Made Easy*. John Wiley Sons Inc., 1 edition, 1998.

[3] Y. Chan and K. Ho. A simple and efficient estimator for hyperbolic location. *IEEE Trans. Signal Processing*, 42:1905–1915, August 1994.

[4] L. Coyle, J. Ye, E. Loureiro, S. Knox, S. Dobson, and P. Nixon. A proposed approach to evaluate the accuracy of tag-based location systems. *Workshop on Ubiquitous Systems Evaluation, UbiComp 2007 Workshop Proceedings*, pages 292–296, 2007.

[5] P. Dana. Global positioning system overview. Technical report, The Geographer's Craft Project, Department of Geography, The University of Colorado at Boulder, http://www.colorado.edu/geography/gcraft/notes/gps/gpsf.html, 1999.

[6] B. Denis, J. Keignart, and N. Daniele. Impact of nlos propagation upon ranging precision in uwb systems. *IEEE Conference on Ultra Wideband Systems and Technologies*, (379-383), Nov 2003.

[7] B. Denis, L. Ouvry, B. Uguen, and F. Tchoffo-Talom. Advanced bayesian filtering techniques for uwb tracking systems in indoor environments. *IEEE International Conference on Ultra-Wideband*, pages 638–643, September 2005.

[8] The Home Depot. *Stanley S2 Laser Level Square*. http://www.homedepot.com.

[9] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez. Particle filtering. *Signal Processing Magazine, IEEE*, 20(5):19–38, Sep 2003.

[10] Ekahau. Comparison of wireless indoor positioning technologies. Whitepaper, Ekahau Inc., http://www.integratedsolutionsmag.com/, 2006.

[11] R. Fontana. Recent system applications of short-pulse ultra-wideband (uwb) technology. *IEEE Microwave Theory and Tech*, 52(9), September 2004.

[12] R. Fontana., E. Richley, and J. Barney. Commercialization of an ultra wideband precision asset location system. *IEEE Conference on Ultra Wideband Systems and Technologies*, pages 369–373, Nov 2003.

[13] C. Gentile and A. Kik. A comprehensive evaluation of indoor ranging using ultra-wideband technology. *EURASIP Journal on Wireless Communications and Networking*, 2007, 2007.

[14] I. Getting. The global positioning system. *IEEE Spectrum*, 30(12):43–47, December 1993.

[15] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios. *IEEE Signal Processing Magazine*, 22(4):70–84, 2005.

[16] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE proceedings F*, 140(2):107–113, April 1993.

[17] W. Guier and G. Weiffenbach. Genesis of satellite navigation. *John Hopkins APL Technical Digest*, 19(1):14–17, 1998.

[18] S. Ingram, D. Harmer, and M. Quinlan. Ultrawideband indoor positioning systems and their use in emergencies. *Position Location and Navigation Symposium, IEEE*, pages 706– 715, April 2004.

[19] K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2:1012–1022, March 2004.

[20] H. Khoury and V. Kamat. Evaluation of position tracking technologies for user localization in indoor construction environments. *Automation in Construction*, 18(4):444–457, July 2009.

[21] M. Kihara. Study of a gps satellite selection policy to improve positioning accuracy. *Position Location and Navigation Symposium, IEEE*, pages 267–273, April 1994.

[22] Ubisense Limited. *Ubisense LocationEngineConfig User Manual*. http://www.ubisense.net, 2.0 edition, 2007.

[23] H. Liu, H. Darabi, P. Banerjee, and L. Jing. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1067–1080, November 2007.

[24] K. Lowe. Distance estimation between transceivers over short distances. Master's thesis, Clemson University, December 2007.

[25] Matlab. *Matlab Documentation*. http://www.mathworks.com/.

[26] P. Maybeck. *Stochastic Models, Estimation and Control*, volume 1 of *Mathematics in Science and Engineering*. New York: Academic Press, 1979.

[27] A. Muqaibel, A. Safaai-Jazi, A. Bayram, A. Attiya, and S. Riad. Ultrawideband through-the-wall propagation. *IEEE Proceedings: Microwaves, Antennas Propagation*, 152(6):581–588, 2005.

[28] R. Scholtz, D. Pozar, and W. Namgoong. Ultra-wideband radio. *EURASIP Journal on Applied Signal Processing*, 3:252–272, 2005.

[29] J. Smith and J. Abel. Closed-form least squares source location estimation from range difference measurements. *IEEE Trans. Acoust. Speech, Signal Processing*, 35:1661–1669, Dec 1987.

[30] R. Torralbas and J. Alvarez. Gps world. Final project for ee 476, Cornell University, http://instruct1.cit.cornell.edu/courses/ee476/.

[31] Ubisense. *Class-Leading Precise Location Fact Sheet*. http://www.ubisense.net/pdf/fact-sheets/products/software/Precise-Location-EN090624.pdf.

[32] Ubisense. *Ubisense Series 7000 Compact Tag Fact Sheet*. http://www.ubisense.net/pdf/fact-sheets/products/hardware/Series-7000-Compact-Tag-EN090624.pdf.

[33] Ubisense. *Ubisense Series 7000 Sensor Fact Sheet*. http://www.ubisense.net/pdf/fact-sheets/products/hardware/Series-7000-Sensor-EN090624.pdf.

[34] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, March 2002.

[35] K. Yu and I. Oppermann. Performance of uwb position estimation based on time-of-arrival measurements. *Joint Conference on Ultrawideband Systems and Technologies. UWBST and IWUWBS.*, pages 400–404, May 2004.

[36] K. Yu and I. Oppermann. Uwb positioning for wireless embedded networks. *IEEE Conference on Radio and Wireless*, pages 459–462, Sep 2004.

[37] C. Zhang, M. Kuhn, B. Merkl, M. Mahfouz, and A. Fathy. Development of an uwb indoor 3d positioning radar with millimeter accuracy. *Microwave Symposium Digest. IEEE MTT-S International*, pages 106–109, June 2006.